Acta de Correcciones al Documento de Trabajo de Grado

Santiago de Cali, 23/07/2024

Autor: Ximena Martinez Ospina; Cristian David Mariaca Rueda

Título del Trabajo de Grado: ANALISIS DE CLUSTERIZACIÓN DE CLIENTES ALERTADOS POR POSIBLES OPERACIONES SOSPECHOSAS EN BANCOLOMBIA

Director: Andres Felipe Cano Cadavid

Como indica el artículo 2.13 de las Directrices para Trabajo de Grado de Maestría, he verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Trabajo de Grado definieron que se efectuaran, como consta en el Acta de Evaluación correspondiente.

_____

Firma del Director del Trabajo de Grado

Santiago de Cali, 7 de junio de 2024

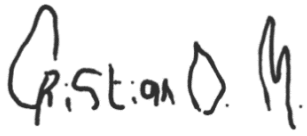
Ingeniero:
Juan Carlos Martínez Arias
Director Posgrados de Ingeniería
Facultad de Ingeniería y Ciencias
Pontificia Universidad Javeriana - Cali

Con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el Trabajo de Grado y posteriormente optar por el título de Magíster en Ciencia de Datos, nos permitimos presentar a su consideración el proyecto de Trabajo de Grado denominado ANALISIS DE CLUSTERIZACIÓN DE CLIENTES ALERTADOS POR POSIBLES OPERACIONES SOSPECHOSAS EN BANCOLOMBIA, el cual será realizado por el (la) estudiante Ximena Martinez Ospina y Cristian David Mariaca Rueda con código 120492289 - 1152452163 perteneciente al énfasis en N/A, bajo la dirección del profesor Andres Felipe Cano Cadavid.

El suscrito director del Trabajo de Grado autoriza para que se proceda a hacer la evaluación de este Proyecto ante el Tribunal que para el efecto se designe, toda vez que ha revisado cuidadosamente el documento y avala que ya se encuentra listo para ser presentado oficialmente.

Atentamente,

Firma
Ximena Martinez Ospina
C.C. 1020492289 de Bello



Firma
Cristian David Mariaca Rueda
C.C. 1152452163 de Medellín

Firma director
Andres Felipe Cano Cadavid
C.C. 71331944 de Medellín

Grupo Bancolombia

07 de junio 2024

Señor:
Juan Carlos Martínez Arias
Director Posgrados de Ingeniería
Pontificia Universidad Javeriana - Cali

Por medio de la presente, me permito dirigirme a usted en calidad de directora de la **Dirección detección e investigación de cumplimiento de Bancolombia S.A**, con el fin de informar que hemos otorgado el permiso necesario a nuestros empleados, **Ximena Martinez Ospina** y **Cristian David Mariaca Rueda**, bajo la dirección de **Andres Felipe Cano Cadavid**, para que realicen su trabajo de grado utilizando información del Banco.

Los estudiantes **Ximena Martinez Ospina** y **Cristian David Mariaca Rueda** están actualmente matriculados en el programa de maestría en ciencia de datos en **Pontificia Universidad Javeriana – Cali** y han solicitado utilizar datos e información de **Bancolombia S.A** para su investigación titulada **"Análisis de clusterización de clientes alertados por posibles operaciones sospechosas en Bancolombia"**. Tras una evaluación interna, hemos decidido apoyar su solicitud bajo los siguientes términos y condiciones:

1. Propósito de la Investigación: Los datos e información proporcionados por Bancolombia serán utilizados exclusivamente para el desarrollo del trabajo de grado de Ximena Martinez Ospina y Cristian David Mariaca Rueda y no podrán ser empleados para otros fines sin nuestro consentimiento expreso.
2. Confidencialidad y Privacidad: Ximena Martinez Ospina y Cristian David Mariaca Rueda se compromete a mantener la confidencialidad de los datos proporcionados, cumpliendo con todas las leyes y regulaciones aplicables en materia de protección de datos. La información no será divulgada a terceros sin la autorización previa de Bancolombia.
3. Supervisión y Contacto: Para cualquier consulta o supervisión relacionada con este permiso, pueden ponerse en contacto con Lina Marcela Monsalve Upegui a través del correo electrónico linamons@bancolombia.com.co

Agradecemos la colaboración de Pontificia Universidad Javeriana - Cali en el desarrollo académico de nuestros empleados y esperamos que esta investigación sea de mutuo beneficio.

Atentamente,

Lina Marcela Monsalve U

Lina Marcela Monsalve Upegui
Directora de cumplimiento
Bancolombia S.

Contact

3116050287 (Mobile)
acanocad@gmail.com

www.linkedin.com/in/andrés-felipe-cano-cadavid-80676115 (LinkedIn)

Top Skills

Liderazgo

Dirección y desarrollo de equipos de trabajo

Ingeniería de datos

Languages

Inglés

Honors-Awards

Ser Ejemplo

Andrés Felipe Cano Cadavid

Líder Área de Conocimiento Analítica de Cumplimiento
Colombia

Summary

Ingeniero con experiencia liderando equipos técnicos expertos en desarrollo de software y analítica; en la sector de educación superior y bancario. Participación en proyecto orientados a la prevención y detección de riesgo, analítica espacial y optimización bajo incertidumbre.

Experience

Bancolombia
10 years 5 months

Líder Área de Conocimiento Analítica Cumplimiento
December 2022 - Present (6 months)
Medellín, Antioquia, Colombia

Responsable del manejo y desarrollo de líderes e integrantes de las líneas conocimiento analítica, ingeniería de datos y ciencia de datos, con el fin de participar en iniciativa que evolucionen las funciones de la Vicepresidencia de Cumplimiento

Líder Línea de Conocimiento Analítica de Cumplimiento
January 2022 - November 2022 (11 months)
Medellín, Antioquia, Colombia

Responsable de potencializar las diferentes funciones de cumplimiento, buscando eficacia en la operación, a partir de la ingeniería de datos y la ciencia de datos; desarrollar el conocimiento analítico en en Centro de Excelencia

Gerente
February 2020 - January 2022 (2 years)
Medellín, Antioquia, Colombia

Responsable de potencializar las diferentes funciones de cumplimiento, buscando eficacia en la operación, a partir de la ingeniería de datos, los modelos analíticos y soluciones tecnológicas.

Jefe de Sección
April 2015 - January 2020 (4 years 10 months)

Medellín, Antioquia, Colombia

Proponer, planear y administrar la gestión analítica y los modelos de cuantificación, scoring y segmentación de factores de riesgos, a través de la inteligencia con la información, en materia de administración de riesgo, del Grupo Bancolombia, de acuerdo con lo definido en el modelo corporativo y analítico, normatividad nacional e internacional y mejores prácticas, que permitan soportar la toma de decisiones comerciales y estratégicas al crear conocimiento a través de la generación de información confiable y analítica, y la gestión adecuada para la mitigación de los riesgos.

Analista
January 2013 - March 2015 (2 years 3 months)

Diseñar, implementar y ejecutar proyectos de inteligencia con la información y gestión analítica a través de metodologías de minería de datos como el análisis predictivos y/o descriptivos, modelos de cuantificación, scoring y técnicas avanzadas de modelamiento, estadísticos y/o matemático.

Tecnológico de Antioquia - Institución Universitaria
Docente de Cátedra
March 2019 - March 2019 (1 month)
Medellín

Docente de posgrado en la Maestría de Gestión de Tecnología

Fundación Universitaria Luis Amigó
Docente de Cátedra
January 2015 - June 2015 (6 months)

Cursos: Base de Datos Avanzadas y Gerencia de Sistemas

Tecnológico de Antioquia - Institución Universitaria
Profesor de Cátedra
February 2012 - June 2014 (2 years 5 months)

Docente de cátedra de pregrado de las asignaturas Probar Software, Métodos Numéricos y Estándares de Calidad

Ruta 3
Director de Desarrollo
May 2012 - January 2013 (9 months)
Medellín, Colombia

Dirigir el área de desarrollo de software, coordinar los proyectos de desarrollo de software.

Corporacion Universitaria Lasallista

Profesor de Cátedra
December 2012 - December 2012 (1 month)

Seminario de fundamentos de Java

Tata Consultancy Services

Analista Funcional
June 2011 - May 2012 (1 year)

Liderar y coordinar la ejecución ordenada de los procesos de especificación, diseño e implementación de soluciones para los requerimientos, proyectos e incidentes que se me encomiendan, garantizar la calidad de las soluciones a cargo, verificando los elementos de calidad en cada entregable y desarrollando mis propios entregables de acuerdo a los procesos y configurar componentes de la plataforma a su cargo teniendo en cuenta aspectos de Arquitectura y Seguridad.

Universidad EAFIT

11 years 5 months

Coordinador Desarrollo de Software
January 2009 - May 2011 (2 years 5 months)

Coordinar los proyectos de desarrollo de software y dar soporte a la plataforma de base de datos y servidor de aplicaciones.

Asistente de Investigación
January 2009 - November 2010 (1 year 11 months)

Asistente de investigación de los proyectos:
- Nuevos métodos para el análisis de datos espaciales: combinación de técnicas esda y técnicas de optimización heurísticas.
- Rediseño del software Risicar para su aplicación en el ámbito empresarial
- Diseño de un software de auditoría integral para uso académico
- Optimización Multi-intervalo valuada y Optimización Simulación
- Agregación de datos espaciales para el análisis económico: Nuevos métodos y aplicaciones

Analista de Sistemas
January 2000 - December 2008 (9 years)

Desarrollar, soportar y mantener los sistemas de información de la Universidad

Education

Universidad EAFIT

Magíster en Matemáticas Aplicadas · (2009 - 2011)

Universidad EAFIT

Ingeniero Matemático · (2004 - 2008)

Universidad EAFIT

Especialista, Desarrollo de Software · (2002 - 2003)

Universidad EAFIT

Ingeniero de Sistemas, Desarrollo de Software · (1995 - 2000)

Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias

FICHA RESUMEN
TRABAJO DE GRADO DE MAESTRÍA

TITULO: "ANALISIS DE CLUSTERIZACIÓN DE CLIENTES ALERTADOS POR POSIBLES OPERACIONES SOSPECHOSAS EN BANCOLOMBIA"

1. ÉNFASIS: N/A
2. TIPO DE PROYECTO: Aplicado
3. ÁREA DE TRABAJO: Riesgo de Lavado de Activos y Financiación del Terrorismo – Sector financiero
4. ESTUDIANTE (S): Ximena Martinez Ospina - Cristian David Mariaca Rueda
5. CORREO ELECTRÓNICO: xime1020na@javerianacali.edu.co, cristiandmr777@javerianacali.edu.co
6. DIRECCIÓN Y TELÉFONO: Carrera 48 #26-85 - 4044000
7. DIRECTOR: Andres Felipe Cano Cadavid
8. VINCULACIÓN DEL DIRECTOR: Externo
9. CORREO ELECTRÓNICO DEL DIRECTOR: acanocad@gmail.com
10. GRUPO O EMPRESA QUE LO AVALA: Bancolombia
11. PALABRAS CLAVE: Lavado de activos, financiación del terrorismo, aprendizaje no supervisado, clusterización, Kmeans, análisis de componentes principales.
12. ODS QUE APLICA EL PROYECTO (Agenda 2030):
13. FECHA DE INICIO: 1/06/2023
14. RESUMEN: En la Vicepresidencia de Cumplimiento en Bancolombia cada mes se generan alertas de clientes con posibles operaciones sospechosas, identificadas a partir de modelos analíticos detectivos. El crecimiento del negocio y aumento de la cobertura de tipologías ha derivado en un aumento de alertas, saturando la capacidad de análisis del área de Investigación, lo que impide generar una respuesta oportuna para mitigar el riesgo de Lavado de Activos y Financiación del Terrorismo (LAFT). Por el aumento de alertas, se han implementado algunos métodos de agrupación que emplean procedimientos intuitivos y requieren aproximadamente tres días hábiles para su ejecución. Para el área de Investigación es útil este proyecto, ya que se centra en buscar la mejora de los procesos de evaluación impactando dos aspectos relevantes a la hora de identificar riesgos LAFT: capacidad y tiempo oportuno de evaluación de las alertas. El objetivo principal del trabajo es implementar modelos de clusterización a partir de técnicas de aprendizaje de máquina para agrupar a los clientes alertados según características de riesgo LAFT que estos representan para el Banco. Además, se busca identificar las variables más relevantes e influyentes en el riesgo LAFT de un cliente alertado. Se espera obtener un modelo de agrupamiento para clientes con posibles operaciones sospechosas en Bancolombia, tener claras las variables, características y patrones que tienen los clientes alertados por operaciones sospechosas, para ser tenidas en cuenta en los monitoreos del Banco, y de esta forma, aportar a que el indicador de oportunidad en el tiempo de respuesta de las alertas sea optimo.

ANÁLISIS DE CLUSTERIZACIÓN DE CLIENTES ALERTADOS POR POSIBLES OPERACIONES SOSPECHOSAS EN BANCOLOMBIA

Ximena Martinez Ospina
Cristian David Mariaca Rueda

Proyecto Aplicado para optar al título de
Magister en Ciencia de Datos

Director
Andrés Felipe Cano Cadavid

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, 202

TABLA DE CONTENIDO

LISTA DE FIGURAS

INTRODUCCIÓN

Bancolombia, actualmente el banco con mayor número de clientes en Colombia tiene la obligación de contar con un sistema de antilavado efectivo. Por ende, cuenta con un área dedicada a detectar alertas en los comportamientos transaccionales de sus clientes, determinar si son sospechosas y reportarlas a la autoridad correspondiente. Debido al crecimiento del negocio, se ha presentado una saturación en la capacidad de análisis del área de investigación de cumplimiento, impactando los tiempos de respuesta oportuna. Para abordar el problema, se implementaron modelos analíticos que permitieron el agrupamiento entre los clientes que presentaron alertas por su comportamiento transaccional, basándose en variables de riesgo LAFT. De esta manera, se evaluaron las señales de alerta de acuerdo con los grupos resultantes de los modelos de clusterización implementados, que cumplen con los estándares de calidad requeridos por el problema. Esto contribuyó a optimizar el indicador de respuesta de las alertas sin afectar la calidad de la investigación.

1. DEFINICIÓN DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En los últimos años, ha aumentado el número de Reportes de Operaciones Sospechosas (ROS) por posible lavado de activos realizados a la UIAF en Colombia [1]. Un ROS es: *"aquella operación que por su número, cantidad o características no se enmarca en el sistema y prácticas normales del negocio, de una industria o de un sector determinado y, además, que de acuerdo con los usos y costumbres de la actividad que se trate, no ha podido ser razonablemente justificada"* [2]. El incremento de ROS es consecuencia del aumento de clientes alertados, resultantes del fortalecimiento estratégico de detección e investigación y del crecimiento del negocio financiero.

En la Vicepresidencia de Cumplimiento en Bancolombia, mensualmente se generan alertas de clientes con posibles operaciones sospechosas, identificadas a partir de modelos analíticos detectivos. El crecimiento del negocio y la ampliación de la cobertura de tipología han derivado en un incremento de alertas, saturando la capacidad de análisis del área de Investigación, lo que impide generar una respuesta oportuna para mitigar el riesgo de Lavado de Activos y Financiación del Terrorismo (LAFT). El riesgo LAFT es: *"la posibilidad de pérdida o daño que puede sufrir una entidad por su propensión a ser utilizada directa o a través de sus operaciones, como instrumento para cometer delitos de LA (Lavado de Activos) o canalización de recursos para la FT (Financiación del Terrorismo)"* [3].

Inicialmente, los clientes alertados se evaluaban de forma individual; sin embargo, con el aumento de alertas, se han implementado algunos métodos de agrupación que emplean procedimientos intuitivos y requieren aproximadamente tres días hábiles para su ejecución, ya que las herramientas utilizadas no son las más adecuadas. Por este motivo, se busca tener un proceso de evaluación más sostenible en el tiempo, explorando técnicas de agrupación asociadas a aprendizaje automático.

1.2. FORMULACIÓN DEL PROBLEMA

- ¿Cómo se desarrolló el modelo de clusterización para formar la mejor agrupación de clientesalertados según el riesgo LAFT que estos representan para el Banco?
- ¿Cuáles fueron las variables más influyentes en el riesgo LAFT que un cliente representa parael Banco?
- ¿Cuáles fueron las técnicas y métricas estadísticas más adecuadas para evaluar el nivel de homogeneidad dentro de cada clúster?
- ¿Qué patrones y características se identificaron en los clientes con posibles operaciones sospechosas?
- ¿La ejecución de un modelo a partir de técnicas de aprendizaje de maquina pudo ser más efectivo que las técnicas actualmente empleadas para la clusterización de clientes alertados?

2. OBJETIVOS DEL PROYECTO

2.1. OBJETIVO GENERAL

Implementar modelos de clusterización a partir de técnicas de aprendizaje de máquina para agrupar a los clientes alertados según características de riesgo LAFT que estos representan para el Banco, permitiendo identificar grupos de clientes alertados que pueden ser evaluados en forma colectiva a través de la misma estrategia, logrando mayor oportunidad en la determinación de la operación sospechosa y mayor eficiencia, disminuyendo el represamiento de alertas.

2.2. OBJETIVOS ESPECÍFICOS

- Identificar las variables más relevantes e influyentes en el riesgo LAFT de un cliente alertado, las cuales serán las variables candidatas para el modelo de clusterización.
- Entrenar modelos de clusterización con diferentes técnicas de aprendizaje no supervisado.
- Validar técnicas estadísticas para evaluar la homogeneidad de los grupos formados por los diferentes modelos ML, para identificar el modelo más adecuado entre los algoritmos entrenados.
- Reconocer características y patrones en los grupos de clientes resultantes de los modelos implementados, que pueden ser relevantes para detectar posibles operaciones sospechosas.
- Emplear técnicas analíticas para mejorar la capacidad de análisis del área de investigación, en lugar de emplear métodos intuitivos que requieren un mayor tiempo de ejecución.

3. MARCO TEÓRICO Y ANTECEDENTES

3.1. MARCO TEÓRICO

En el desarrollo de la gestión de riesgos LAFT (investigación, evaluación y mitigación), es necesario conocer el significado de tipología, uno de los conceptos más importantes que permiten generar señales de alerta efectivas y focalizar la implementación de los monitoreos.

El Grupo de Acción Financiera Internacional (GAFI), institución intergubernamental cuyo propósito es desarrollar políticas que ayuden a combatir el lavado de activos y la financiación del terrorismo, define las tipologías como aquellos modus operandi utilizados por las organizaciones criminales para la colocación, ocultamiento e integración del dinero producto de las actividades ilícitas o lícitas realizadas por estas. Cuando una serie de hechos o acciones parecen estar edificados de manera estructurada y utilizan los mismos métodos, pueden ser clasificados como herramientas para la identificación de señales de alerta [4].

Las señales son situaciones cuya ocurrencia y detección hacen necesario el análisis adicional en busca de posibles explicaciones para los hechos que han llamado la atención, facilitando el reconocimiento en este contexto de una posible operación de lavado de activos o

financiación del terrorismo. Una señal de alerta debe entenderse de manera individual y no concurrente ni bajo el supuesto de que la detección de una operación de lavado de activos depende de la presentación de una, algunas, o todas las señales de alerta [5]. No obstante, con aprendizaje automático, se busca obtener un modelo de clusterización que permita la división de objetos similares en conjuntos de acuerdo con un grupo de características comunes, en este caso clientes que presentan señales de alerta. En efecto, para el desarrollo de los modelos de clusterización, la librería de Python sklearn es altamente utilizada y muy popular, proporcionando una amplia gama de algoritmos de clustering. Otra librería importante para facilitar la elección del modelo más adecuado es Pycaret, que automatiza los workflows de Machine Learning.

Para determinar qué métodos de agrupamiento implementar, se investigaron y seleccionaron los algoritmos más populares como: K-Means, DBSCAN, Affinity Propagation, Mean Shift, Agglomerative Clustering, OPTICS, BIRCH, Spectral Clustering. También nos basamos en el conocimiento experto en segmentación LAFT del área de analítica de la Vicepresidencia de Cumplimiento, enfocándonos en los resultados obtenidos en la segmentación SARLAFT de todos los clientes del Banco, en donde los algoritmos que mejor presentaron resultados fueron MiniBatch K-Means y Spectral Clustering.

A continuación, se expone el funcionamiento de las librerías y las técnicas empleadas:

Pycaret

PyCaret es una librería de Python que automatiza flujos de trabajo de aprendizaje automático, permitiendo llevar a cabo desde la preparación de los datos hasta el despliegue del modelo final en tan solo unos minutos. También permite realizar comparaciones de varios modelos automáticamente, con el fin de facilitar la elección del mejor modelo para el problema. Esta herramienta acelera exponencialmente el ciclo experimental del modelado de datos y lo hace más productivo. Una ventaja de PyCaret es que se puede utilizar para reemplazar cientos de líneas de código con solo unas pocas líneas.

PyCaret es esencialmente un contenedor de Python para varias bibliotecas como scikit-learn, XGBoost, LightGBM, CatBoost, Optuna, Hyperopt, entre otros. Esta librería se inspiró en la biblioteca caret del lenguaje de programación R.

Scikit-Learn (Sklearn)

Sklearn es una librería de Python que cuenta con algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad. Esta herramienta cuenta con varios componentes:

- Algoritmos de aprendizaje automático.
- Herramientas de procesamiento de datos.
- Herramientas para la evaluación de modelos.
- Herramientas de selección de características.
- Herramientas de ajuste de modelos.

La librería cuenta con múltiples ventajas, de las cuales hay cuatro muy relevantes:

- La sintaxis que emplea la biblioteca es igual para todos los modelos.
- Está basada en NumPy, SciPy y matplotlib, lo que permite beneficiarse del código que utilizan estas librerías.
- La biblioteca soporta algoritmos de última generación como KNN, XGBoost, bosque aleatorio, SVM, entre otros.
- Los algoritmos de Scikit-Learn se combinan y depuran con otras estructuras de datos y aplicaciones externas como Pandas o PyBrain, que son sencillas de comprender.

La gran variedad de algoritmos y utilidades de Sklearn la convierten en la herramienta básica para empezar a programar y estructurar los sistemas de análisis de datos y modelado estadístico, pues proporciona una variedad de módulos y algoritmos que facilitan el aprendizaje y trabajo del científico de datos en las primeras fases de su desarrollo.

K-Means

K-Means es un algoritmo de clasificación no supervisado (clusterización) que agrupa individuos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada individuo y el centroide de su grupo o clúster.

El algoritmo funciona de la siguiente manera:

1. Inicialmente se especifica el número de clústeres k. Para esto se pueden emplear técnicas como el método del codo o validación cruzada.
2. Posteriormente, se eligen k puntos al azar del conjunto de datos de donde se toman los centroides iniciales de cada clúster.
3. Luego, el algoritmo asigna cada punto del conjunto de datos al clúster cuyo centroide esté más cerca. Esto se hace calculando la distancia entre cada punto y cada centroide, y se asigna el punto al clúster cuyo centroide tenga la menor distancia.
4. Cuando todos los puntos han sido asignados a un clúster, se recalculan los centroides de cada clúster como la media de todos los puntos del clúster; es decir, se actualiza la posición del centroide para reflejar la nueva agrupación.
5. Se repiten los pasos 3 y 4 hasta que se presente alguno de estos tres escenarios: los centroides dejan de cambiar, los puntos dejan de cambiar de clúster o se alcanza el límite de iteraciones.

El algoritmo básicamente resuelve un problema de optimización, siendo la función para optimizar (minimizar) la suma de las distancias cuadráticas de cada objeto al centroide de su grupo.

Los objetos se representan con vectores reales de n dimensiones $(x_1, x_2, x_3, \dots, x_n)$, y el algoritmo construye k grupos, donde se minimiza la suma de la distancia de los objetos, dentro de cada grupo $\boldsymbol{S} = \{S_1, S_2, S_3, \dots, S_k\}$, a su centroide, así:

$$\min_{\boldsymbol{S}} E(\boldsymbol{\mu_i}) = \min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{\boldsymbol{x_j} \in \mathbf{S_i}} ||\boldsymbol{x_j} - \boldsymbol{\mu_i}||^2 \quad (1)$$

Donde:

- $\boldsymbol{S}$**:** es el conjunto de datos.
- $\boldsymbol{x_j}$**:** Características o atributos.
- $\boldsymbol{k}$**:** Grupos o clústeres
- $\boldsymbol{\mu_i}$**:** Centroides.

En cada actualización de los centroides, Se impone la condición necesaria de extremo a la función $\boldsymbol{E(\mu_i)}$ que, para la función (1) es:

$$\frac{\partial E}{\partial \boldsymbol{\mu_i}} = 0 \rightarrow \boldsymbol{\mu_i}^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{\boldsymbol{x_j} \in \boldsymbol{S_i}^{(t)}} \boldsymbol{x_j}$$

Finalmente, para evaluar la calidad de las agrupaciones realizadas por el algoritmo, se puede recurrir a métricas como la inercia o la puntuación de la silueta.

Algunas de las principales ventajas de emplear K-Means son [6]:

- Es una técnica simple de implementar.
- Se adapta a grandes conjuntos de datos.
- Se adapta a conjuntos de datos con diferentes características.
- Garantiza la convergencia.
- Se generaliza a grupos de diferentes formas y tamaños, como grupos elípticos.

Esta técnica es muy popular en aplicaciones relacionadas con la segmentación de clientes, clasificación de texto o detección de anomalías.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN es un algoritmo de clasificación no supervisado (clusterización) propuesto por Ester, Kriegel, Sander y Xu en 1996, y se basa en densidad. Generalmente, la métrica empleada en este algoritmo es la distancia euclidiana. Este método se caracteriza por emplear dos parámetros:

- **Número mínimo de puntos (> 0)**: Es el número mínimo de puntos requeridos para formar una región densa. La elección acertada de este parámetro permite eliminar correctamente los datos atípicos. Una forma de seleccionar este valor es asegurarse de que sea al menos igual a la dimensionalidad del conjunto de datos. Si se trabaja con un conjunto de datos ruidoso, se debe elegir un valor mayor para el número mínimo de puntos.
- $\boldsymbol{\varepsilon}$ **> 0:** Especifica la proximidad requerida entre puntos para ser considerados parte de un clúster. Es decir, si la distancia entre dos puntos es menor o igual a este valor $\boldsymbol{\varepsilon}$, se consideran vecinos y forman parte del mismo clúster. Una forma de calcular este valor es determinar las distancias entre los puntos más cercanos en una matriz de puntos.

En este método hay 3 tipos de datos:

- Punto núcleo: un punto de núcleo tiene más de un número especificado de puntos mínimos dentro de un radio de $\boldsymbol{\varepsilon}$ a su alrededor. Este punto siempre pertenece a una

región densa.

- Punto borde: un punto de borde tiene menos de puntos mínimos dentro de $\boldsymbol{\varepsilon}$, pero se encuentra en la vecindad de un punto de núcleo.
- Punto ruido: un punto de ruido es cualquier punto que no sea un punto de núcleo o un punto de borde.

Otros dos conceptos que intervienen en la construcción del algoritmo son los siguientes [7]:

- Borde de densidad: Si p y q son puntos centrales y la distancia entre p y q es menor o igual a $\boldsymbol{\varepsilon}$ entonces podemos conectar el vértice de p y q en un gráfico y llamarlo borde de densidad.
- Densidad de puntos conectados: Se dice que dos puntos p y q son puntos de conexión de densidad si tanto p como q son puntos de núcleo y existen una trayectoria formada por los bordes de densidad que conectan el p con el punto q.

El algoritmo funciona de la siguiente manera:

1. Inicialmente se selecciona un punto arbitrario que no ha sido visitado y la información de su vecindario se recupera desde el parámetro $\boldsymbol{\varepsilon}$.
2. En este paso se inicia la formación de clústeres, siempre y cuando el punto seleccionado contenga puntos mínimos dentro del vecindario $\boldsymbol{\varepsilon}$, si esto no se cumple, el punto se etiqueta como ruido. Posteriormente este mismo punto se puede encontrar dentro de la vecindad $\boldsymbol{\varepsilon}$, de un punto diferente y puede ser parte del clúster.
3. Cuando se encuentra un punto de núcleo, los otros puntos dentro del vecindario $\boldsymbol{\varepsilon}$ también son parte del grupo, por tanto, todos los puntos que se encuentran dentro del vecindario se agregan, junto con su propio vecindario $\boldsymbol{\varepsilon}$, si son puntos de núcleo.
4. El proceso concluye cuando se encuentra completamente el clúster conectado a la densidad.
5. El proceso reinicia con la selección de un nuevo punto.

Básicamente, el algoritmo se basa en la detección de áreas donde existen concentraciones de puntos separadas por áreas vacías o con escasos puntos. Los puntos que no forman parte de un clúster se etiquetan como ruido [9].

Algunas de las principales ventajas de emplear DBSCAN son:

- La capacidad de crear un número desconocido de clases.
- Puede crear clases no convexas.
- Es muy útil para detectar anomalías.
- Funciona bien en conjuntos de datos con ruido y valores atípicos.
- A diferencia de otros algoritmos de clusterización, DBSCAN no requiere que el usuario especifique el número de clústeres que se desean generar.
- Es excelente para separar clústeres de alta densidad frente a clústeres de baja densidad.
- Los parámetros son fáciles de entender, lo que facilita que los usuarios puedan elegir los parámetros óptimos de forma más intuitiva [9].

Esta técnica es ampliamente empleada en el sector corporativo, pues sirve para detectar

fraudes, ataques de seguridad, problemas en el tratamiento de datos, en medicina o para realizar mantenimiento preventivo [10].

Spectral Clustering

Spectral Clustering es un algoritmo de clasificación no supervisado (clusterización) basado en teoría de grafos. Esta técnica emplea los valores y vectores propios de los datos para proyectar los datos en un espacio de dimensiones inferiores y agruparlos [23]. Hay tres pasos principales en la clusterización por este método: construir un grafo de similitud, proyectar los datos en un espacio de dimensiones inferiores y agrupar los datos. El grafo de similitud se puede construir de la siguiente manera [11]:

- Totalmente conectado: cada nodo está conectado a todos los otros nodos, y los bordes del grafo están ponderados por la similitud de sus puntos finales.
- K-Vecinos más cercanos: cada nodo está conectado solo a sus k nodos más cercanos.
- $\boldsymbol{\varepsilon}$ – vecindario: los nodos están conectados solo si su distancia es menor que $\boldsymbol{\varepsilon}$.

El enfoque que sigue el algoritmo se basa en la conectividad; es decir, los puntos que están conectados o inmediatamente adyacentes se colocan en el mismo grupo. Incluso si la distancia entre dos puntos es pequeña, si no están conectados, no se agrupan [12].

Dado un conjunto de puntos S en un espacio de dimensiones superiores el algoritmo funciona de la siguiente manera [13]:

1. Inicialmente se genera una matriz de distancias.
2. Luego, esta matriz se transforma en una matriz de afinidad A (grafico de similitud).
3. Posteriormente se calcula la matriz de grados D y la matriz laplaciana $L = D - A$.
4. Se encuentran los valores y vectores propios de la matriz laplaciana.
5. Con los vectores y valores propios más grandes calculados anteriormente, se forma una matriz.
6. Se normalizan los vectores.
7. Por último, se agrupan los puntos de datos en un espacio K-dimensional.

Algunas de las principales ventajas de emplear Spectral Clustering son:

- Los grupos no siguen una forma o patrón fijo; es decir, el algoritmo es eficaz para datos de diferentes formas y tamaños [13].
- El algoritmo es computacionalmente rápido para conjuntos de datos de gran tamaño [13].
- Esta técnica de agrupación, a diferencia de otras técnicas tradicionales, no presupone ninguna propiedad específica de los datos [14].
- El algoritmo utiliza la descomposición de valores propios para reducir la dimensionalidad de los datos, lo que facilita su visualización y análisis [14].
- Es sensible al ruido y a los valores atípicos [14].

Esta técnica es ampliamente empleada para la segmentación de imágenes, extracción de datos educativos, resolución de entidades y agrupación espectral de secuencias de proteínas, entre otros casos [13].

Affinity Propagation

Affinity Propagation es un algoritmo de clasificación no supervisado (clusterización) desarrollado en 2007 por Brendan Frey y Delbert Dueck. Esta técnica destaca por su capacidad para descubrir patrones ocultos en los datos. El algoritmo funciona mediante un proceso de paso de mensajes entre puntos de datos, alternando entre dos tipos de mensajes [15]:

- Responsabilidad: Estos mensajes indican qué tan adecuado es un punto de datos para ser un ejemplo de otro punto.
- Disponibilidad: Estos mensajes muestran la evidencia acumulada de que un punto de datos en particular debería elegir otro punto como su ejemplo.

Estos mensajes se actualizan iterativamente hasta lograr la convergencia.

El algoritmo funciona de la siguiente manera [16]:

1. Inicialmente se hace el cálculo de la matriz de similitud. La elección de la métrica empleada en este paso depende de los datos y del problema que se esté trabajando.
2. Luego, el algoritmo inicializa la matriz de responsabilidad ($\boldsymbol{R}$), donde $\boldsymbol{R}(\boldsymbol{i},\boldsymbol{k})$ representa la responsabilidad del punto de datos $\boldsymbol{i}$ para que sea el ejemplo del punto de datos $\boldsymbol{k}$.
3. De forma similar, el algoritmo inicializa la matriz de disponibilidad ($\boldsymbol{A}$), donde $\boldsymbol{A}(\boldsymbol{i},\boldsymbol{k})$ representa la disponibilidad del punto de datos $\boldsymbol{i}$ como su ejemplo.
4. Una vez se inicializan las matrices de los pasos 2 y 3, se actualizan de forma iterativa hasta la convergencia que se alcanza cuando las matrices ya no cambian significativamente entre iteraciones.
5. Posteriormente se hace el cálculo de la responsabilidad neta su mando la responsabilidad y disponibilidad de cada punto.
6. Se identifican ejemplos como puntos de datos con alta responsabilidad neta.
7. Finalmente asigna cada punto de datos al ejemplar más cercano para formar grupos.

Algunas de las principales ventajas de emplear Affinity Propagation son:

- No requiere que el usuario especifique la cantidad de clústeres.
- Tiene una alta capacidad para manejar distribuciones de datos complejas y no lineales.
- Se puede aplicar a grandes conjuntos de datos, aunque puede requerir optimización para lograr eficiencia computacional [16].

Esta técnica es ampliamente empleada en segmentación de imágenes, análisis de expresión genética, agrupación de documentos, análisis de redes sociales, segmentación de mercado, entre otros [15].

Meanshift

Meanshift es un algoritmo de clasificación no supervisado (clusterización) basado en centroides, que funciona actualizando iterativamente los candidatos a centroides para que sean la media de los puntos dentro de una región determinada. Posteriormente, estos candidatos se filtran en una etapa de postprocesamiento para formar el conjunto final de centroides [17]. El nombre de la técnica deriva del hecho de que en cada iteración del algoritmo hay un "desplazamiento de la

media".

Este algoritmo cuenta con un parámetro conocido como ancho de banda. Dependiendo del ancho de banda, los clústeres pueden tener aspectos bastante diferentes. Elegir manualmente el ancho de banda adecuado puede funcionar para conjuntos de datos pequeños y bidimensionales, pero puede ser bastante difícil a medida que el conjunto de datos crece [17]. Sin embargo, existen técnicas que pueden ayudar en la inferencia de este parámetro.

El algoritmo funciona de la siguiente manera [18]:

1. Inicialmente se estima la función de densidad de probabilidad subyacente de los puntos de datos. Esto generalmente se hace mediante la estimación de la densidad del núcleo, donde cada punto de datos está representado por una función del núcleo centrada en ese punto.
2. La función kernel especifica el peso asignado a cada punto de datos en el proceso de estimación de densidad.
3. Luego, el algoritmo desplaza iterativamente los puntos de datos hacia regiones de mayor densidad. El cambio se determina calculando el vector de cambio medio para cada punto de datos, que representa la dirección y magnitud del cambio. El vector de desplazamiento medio se calcula como el promedio ponderado de las diferencias entre el punto de datos y sus puntos vecinos, donde los pesos están determinados por la función kernel.
4. El algoritmo continúa desplazando los puntos de datos hasta que se alcanza la convergencia. La convergencia ocurre cuando los vectores de desplazamiento medios se vuelven muy pequeños o insignificantes.
5. Una vez que se logra la convergencia, la posición final de cada punto de datos representa un centro de clúster.
6. Por último, el algoritmo asigna cada punto de datos al centro del grupo más cercano, identificando así los grupos dentro de los datos.

Algunas de las principales ventajas de emplear Meanshift son:

- No requiere que el usuario especifique la cantidad de clústeres.
- Tiene una alta capacidad para manejar distribuciones de datos complejas y no lineales.
- El algoritmo no es sensible a los valores atípicos y la convergencia está garantizada.

Esta técnica es muy empleada en la segmentación de imágenes, en áreas como imágenes médicas, conducción automatizada, videovigilancia, reconocimiento y detección facial, procesamiento de imágenes satelitales, entre otras aplicaciones de visión artificial [19].

Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering es un algoritmo de clasificación no supervisado (clusterización) y es uno de los métodos jerárquicos más comunes para agrupar datos. Los clústeres resultantes suelen ser representados mediante un dendrograma. La técnica funciona de manera "ascendente", donde inicialmente cada elemento se considera como un grupo de un solo elemento y en cada paso del algoritmo, los dos grupos más similares se combinan en un nuevo grupo más grande. La iteración termina cuando todos los puntos son miembros de un

solo grupo [20].

El algoritmo funciona de la siguiente manera [21]:

1. Inicialmente se calcula la matriz de proximidad. Para determinar la similitud entre los datos se pueden emplear diferentes enfoques como: mínimo, máximo, promedio del grupo, distancia entre centroides, enlace simple, enlace flexible, vinculación completa, vínculo promedio de grupos de pares no ponderados, vínculo promedio ponderado par-grupo, método Ward, entre otros.
2. Cada punto se considera inicialmente como un grupo de un solo elemento.
3. Se fusionan iterativamente los dos grupos más cercanos y se actualiza la matriz de proximidad.
4. Se repite el proceso hasta que quede un único grupo que contenga todos los puntos.

Algunas de las principales ventajas de emplear Agglomerative Hierarchical Clustering son:

- No requiere que el usuario especifique la cantidad de clústeres.
- Puede representar taxonomías significativas.

OPTICS

OPTICS es un algoritmo de clasificación no supervisado (clusterización) que significa ordenar puntos para identificar la estructura del clúster. Este algoritmo está inspirado en DBSCAN y fue desarrollado por los mismos creadores. El objetivo principal de esta técnica es extraer la estructura de agrupamiento de un conjunto de datos mediante la identificación de puntos conectados por densidad [22].

Uno de los parámetros que se debe configurar es el valor $\boldsymbol{\varepsilon}$ máximo, que determina la distancia máxima entre dos puntos para que sean considerados parte del mismo grupo. El valor ε máximo óptimo se puede determinar mediante el gráfico de accesibilidad que genera el algoritmo OPTICS. Este gráfico muestra la distancia entre cada punto y su vecino más cercano que tiene una mayor densidad [23].

El algoritmo funciona de la siguiente manera [22]:

1. Inicialmente se calcula el valor de $\boldsymbol{\varepsilon}$.
2. Para cada punto se calcula la distancia a sus K-vecinos más cercanos.
3. A partir de un punto arbitrario se calcula la distancia de accesibilidad de cada punto del conjunto de datos, en función de la densidad de sus vecinos.
4. Se ordenan los puntos según su distancia de accesibilidad y se crea el grafico de accesibilidad.
5. Por último se extraen conglomerados del grafico de accesibilidad agrupando los puntos que estén cerca entre sí y que tengan distancias similares.

Las nociones de DBSCAN son válidas en este método; sin embargo, hay otros dos conceptos relevantes en esta técnica:

- **Distancia al núcleo:** es el valor mínimo de radio requerido para clasificar un punto

determinado como punto central.

- **Distancia de alcanzabilidad:** la distancia de accesibilidad entre un punto p y q. Tenga en cuenta que la distancia de alcanzabilidad no está definida si q no es un punto central.

Algunas de las principales ventajas de emplear OPTICS son:

- Es útil para identificar grupos de diferentes densidades en conjuntos de datos grandes y de alta dimensión [22].
- Este algoritmo posee una ventaja significativa sobre DBSCAN, ya que puede detectar grupos significativos en datos con densidad variable [24].
- No requiere que el usuario especifique la cantidad de clústeres. Tiene una alta capacidad para manejar distribuciones de datos complejas y no lineales [24].
- Es sensible al ruido y a los valores atípicos [24].

Esta técnica es muy empleada para la segmentación de imágenes, detección de anomalías, segmentación de clientes, detección de fraude, análisis de redes sociales, análisis de imágenes médicas, agrupación de texto, sistemas de recomendación y bioinformática [25].

BIRCH

BIRCH es un algoritmo de clasificación no supervisado (clusterización) propuesto por Tian Zhang, Raghu Ramakrishnan y Miron Livny en 1996 [26]. Su nombre significa "Reducción y Agrupamiento Iterativo Equilibrado utilizando Jerarquías". Por medio de este algoritmo se pueden agrupar grandes conjuntos de datos, produciendo inicialmente un breve resumen que preserve la mayor cantidad de información posible sobre el conjunto de datos, para luego agrupar este resumen en lugar de hacerlo en todo el conjunto de datos. BIRCH posee una limitación, ya que solo considera atributos métricos, es decir, no admite datos categóricos [27].

El algoritmo de agrupamiento BIRCH consta de dos etapas [28]:

- **Creación del árbol CF:** esta técnica resume grandes conjuntos de datos en regiones más pequeñas y densas llamadas entradas de características de agrupación (CF). Cada entrada CF puede estar compuesta por otras entradas CF.
- **Agrupación global:** el método aplica un algoritmo de agrupación existente en las hojas del árbol CF. Un árbol CF es una estructura donde cada nodo hoja contiene un subgrupo. Cada entrada en un árbol CF contiene un puntero a un nodo secundario y una entrada CF formada por la suma de las entradas CF en los nodos secundarios.

El algoritmo funciona de la siguiente manera [28]:

- Se crea un árbol de funciones de agrupación.
- Esta etapa opcional permite condensar en un rango deseable construyendo un árbol CF más pequeño.
- Se realiza la agrupación global.
- Esta etapa opcional permite el refinamiento de los clústeres.

Algunas de las principales ventajas de emplear BIRCH son [28]:

- Por medio de modificaciones, también se puede utilizar para acelerar la agrupación de k-

medias y el modelado de mezclas gaussianas.

- Es capaz de agrupar de forma incremental y dinámica puntos de datos métricos multidimensionales entrantes para producir una agrupación de la mejor calidad.
- Es útil para para realizar una agrupación precisa en conjuntos de datos grandes.

Esta técnica es muy empleada para agrupación de secuencias de genes, agrupación de variantes en función de sus características, genética de poblaciones, farmacogenómica y genómica evolutiva [26].

Prueba de permutación multivariada

La prueba de permutación multivariada es una prueba estadística no paramétrica utilizada para evaluar hipótesis en situaciones donde se tienen múltiples variables dependientes. Las hipótesis a contrastar son las siguientes:

H_0: No hay diferencias significativas entre los grupos
H_1: Hay diferencias significativas entre los grupos

Si el valor-p obtenido es menor o igual que el nivel de significancia, se rechaza la hipótesis nula. El valor-p es la proporción de estadísticos de prueba permutados que son al menos tan extremos como el estadístico observado.

El algoritmo funciona de la siguiente manera:

1. Se calcula un estadístico de prueba, que puede ser alguna medida de tendencia central, medida de dispersión, medida de distancia, estadístico de prueba multivariante, entre otros.
2. Está basado en el reordenamiento aleatorio de los datos; por lo tanto, las etiquetas de los grupos se reordenan aleatoriamente un gran número de veces y se recalcula el estadístico de prueba.
3. Se construye una distribución del estadístico de prueba basada en las permutaciones.
4. Por último, se calcula el valor-p que permite tomar una decisión sobre la hipótesis nula planteada.

Algunas de las principales ventajas de esta técnica son:

- Al ser una prueba no paramétrica, no se requiere de ningún tipo de supuestos sobre los datos.
- Es una prueba robusta para el análisis de datos complejos.
- Es una prueba multivariada, por tanto, puede aplicarse a observaciones medidas en varias variables.

Esta prueba es empleada para evaluar diferencias multivariadas entre grupos.

3.2. ANTECEDENTES

Un proyecto similar a este corresponde a Aitor Zaragoza Galiana, quien en su trabajo titulado "Clustering y Analítica de clientes de SEMIC mediante Machine Learning" (2021), analiza las principales causas por las cuales los clientes de una empresa se sienten insatisfechos con los productos ofrecidos, lo cual reduce su participación con la compañía.

El objetivo de este trabajo es proporcionar a la empresa una herramienta que le permita tener un panorama más amplio de los motivos y causas de la insatisfacción en los clientes. Para lograr esto, se empleó el algoritmo de Machine Learning K-Means para agrupar a los clientes según su similitud, considerando algunas variables que miden el nivel de satisfacción de los clientes. Posteriormente, utilizando una herramienta de visualización, se observaron las características y patrones comunes entre los clientes de cada grupo. Una vez procesada toda la información, se identificaron grupos de clientes con niveles elevados tanto de satisfacción como de insatisfacción. Finalmente, se concluyó que un producto específico podría estar generando insatisfacción entre los clientes, y se recomendó realizar un análisis más profundo a nivel de negocio para tomar medidas respecto a este producto.

Además de este proyecto mencionado, que guarda similitudes con el modelo que se desea implementar en el área de Investigación de la Vicepresidencia de Cumplimiento en Bancolombia, se han implementado métodos más intuitivos para agrupar a los clientes alertados, segmentándolos según ciertas características de riesgo LAFT. Esta segmentación se realiza exclusivamente utilizando Excel, empleando las fórmulas disponibles en esta herramienta.

En la primera etapa de este proceso, se excluyen algunos clientes que, debido a características particulares, deben ser evaluados de manera individual. También se realiza un análisis mediante gráficos para identificar qué clientes pueden ser evaluados en grupos separados debido a datos compartidos que los relacionan.

La segunda etapa comienza con la división de los clientes según el tipo de monitoreo: un grupo de clientes monitoreados por operaciones en moneda nacional y otro grupo monitoreado por operaciones en moneda internacional. En la última etapa, cada grupo se subdivide aún más según el tipo de monitoreo interno, aplicando reglas específicas que permiten agrupar a los clientes en diferentes clústeres. Dado el volumen de clientes y la cantidad de reglas empleadas, así como las limitaciones de Excel para procesar grandes volúmenes de datos, este proceso puede demorar aproximadamente 3 días en ejecutarse.

Actualmente, en la Vicepresidencia de Cumplimiento se está implementando un modelo de segmentación SARLFAT para los clientes de Bancolombia, el cual permite evaluar la situación del banco desde diversas perspectivas. La segmentación es el proceso mediante el cual se separan elementos en grupos homogéneos internamente y heterogéneos entre sí, basándose en diferencias significativas en sus características (variables de segmentación).

La segmentación SARLAFT es un **mecanismo de prevención y control del riesgo ante el lavado de activos y financiación del terrorismo**. Utiliza variables que abarcan los cuatro factores de riesgo LAFT para implementar modelos analíticos que agrupan a los clientes con características similares. Esto permite generar alertas por comportamientos atípicos de los clientes en comparación con sus pares, desde diferentes perspectivas, y gestionar así el riesgo de manera efectiva.

4. IDENTIFICACIÓN DE LAS VARIABLES MÁS RELEVANTES E INFLUYENTES EN EL RIESGO LAFT DE UN CLIENTE ALERTADO

A continuación, se abordarán las etapas realizadas para alcanzar el primer objetivo específico, que comprende la identificación de las variables más relevantes e influyentes en el riesgo LAFT de un cliente alertado. Estas variables serán consideradas como las candidatas para el modelo de clusterización.

4.1. Requisitos de datos

El objetivo del modelo era realizar una segmentación por características homogéneas que indiquen riesgo LAFT entre los clientes alertados. Es fundamental contar con datos precisos, exactos, vigentes y actualizados. Por ello, se optó por utilizar datos transaccionales que cumplen con estas condiciones, ya que son resultado de la operación de los clientes y contienen información cuantitativa. Además, se utilizaron datos generados por la Vicepresidencia de Cumplimiento derivados de la mitigación y gestión de riesgos LAFT. También se cuenta con información sociodemográfica actualizada, suministrada por el cliente a través de la fuerza comercial, en cumplimiento del numeral 4.2.2.2.1.6.1 de la Circular Básica Jurídica (C.E. 029/14), Parte I, Título IV, Capítulo IV, que establece que "las entidades vigiladas pueden definir la periodicidad con la cual se debe realizar la actualización de estos datos que, en todo caso, no puede ser superior a tres años". Para garantizar una cobertura completa de esta etapa, se realizó la identificación del formato de los datos previamente, lo que permitió determinar el tipo de procesamiento necesario para cada uno de ellos. En su mayoría, los datos eran de tipo numérico y texto, con algunos valores flotantes; en menor medida, se encontraron datos de tipo fecha y binarios.

4.2. Recopilación de datos

Esta etapa comenzó con la comprensión de los monitoreos y procesos de investigación que la entidad utiliza para identificar clientes sospechosos de actividades relacionadas con el lavado de activos y financiación del terrorismo. Se realizó un análisis de los factores de riesgo más influyentes, lo cual permitió conocer las variables y modelos empleados para la detección de posibles operaciones sospechosas. Posteriormente, se identificaron los elementos más relevantes e importantes del proceso de investigación para determinar el nivel de riesgo LAFT asociado a un cliente alertado por una operación inusual. De esta manera, se seleccionaron estos elementos para crear el primer conjunto de variables candidatas para el desarrollo del proyecto.

La recolección de datos se basó principalmente en las bases de clientes alertados que se generan mensualmente; estas bases fueron suministradas por la célula de monitoreos, el área de investigación encargada de analizar a los clientes que presentan alertas derivadas de los monitoreos transaccionales. Estos monitoreos son modelos analíticos enfocados en la detección de posibles operaciones sospechosas debido a comportamientos transaccionales atípicos. Se consolidaron todas las alertas generadas durante los periodos de enero a junio de 2023 (6 meses). En esta base se incluían muchas de las variables candidatas.

Para identificar las fuentes de información de las variables restantes, se consultó a expertos sobre

las zonas y tablas específicas del data lake de la organización donde se podían encontrar estas variables. El sistema core bancario, que gestiona y soporta las funciones claves del negocio (plataforma backend), captura y almacena la información en un data lake para su posterior procesamiento y transformación. Cada área propietaria de los datos es responsable de almacenar y garantizar su calidad, cumpliendo así con un buen gobierno de la información. Además, se elaboró una lista de variables que no estaban disponibles pero que podrían ser calculadas a partir de otras.

Una vez identificados los clientes alertados y las variables faltantes, así como su ubicación en las zonas y tablas del data lake, se procedió a la construcción de consultas SQL según las necesidades identificadas. Esto permitió la extracción del conjunto de datos estructurados.

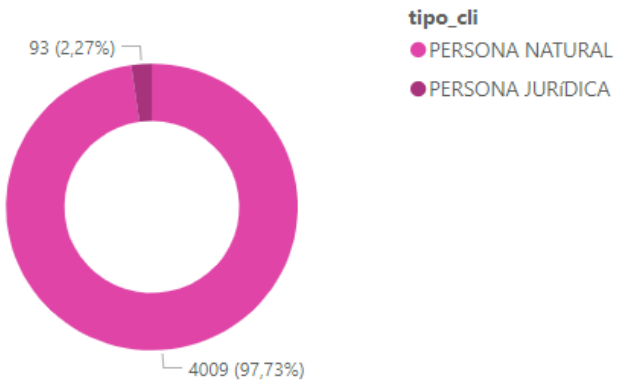
4.3. Comprensión de los datos

Para comprender los datos, fue fundamental entender el proceso de investigación LAFT, realizado en la etapa 2. Esto permitió enfocar el análisis de variables en este contexto, evaluando la calidad y completitud de los datos.
Durante esta fase, se llevó a cabo una exploración preliminar de todas las variables candidatas. Se utilizó estadística descriptiva, agrupación de variables y un tablero en Power BI para identificar patrones y comportamientos de los clientes alertados. A continuación, se destacan los aspectos más relevantes:

- Aproximadamente el 98% de los clientes alertados por monitoreos transaccionales durante el primer semestre de 2023 fueron personas naturales lo cual se puede evidenciar en la figura 1.

Figura 1

Tipo de clientes alertados por monitoreos transaccionales durante el primer semestre del 2023



- De acuerdo con la figura 2 se pudo ver que los segmentos comerciales con mayor representación son el segmento "personal", conformado por un grupo de personas naturales que manejan bajos montos de recursos, y el segmento "independientes", compuesto por personas naturales con un manejo de recursos mucho más alto. Esta concentración se refleja en el gráfico de subsegmentos de la figura 3, donde la mayoría pertenecen a los subsegmentos mediano, alto y pequeño, lo cual es coherente con la distribución del segmento.

Figura 2

Segmentos de los clientes alertados por monitoreos transaccionales durante el primer semestre del 2023

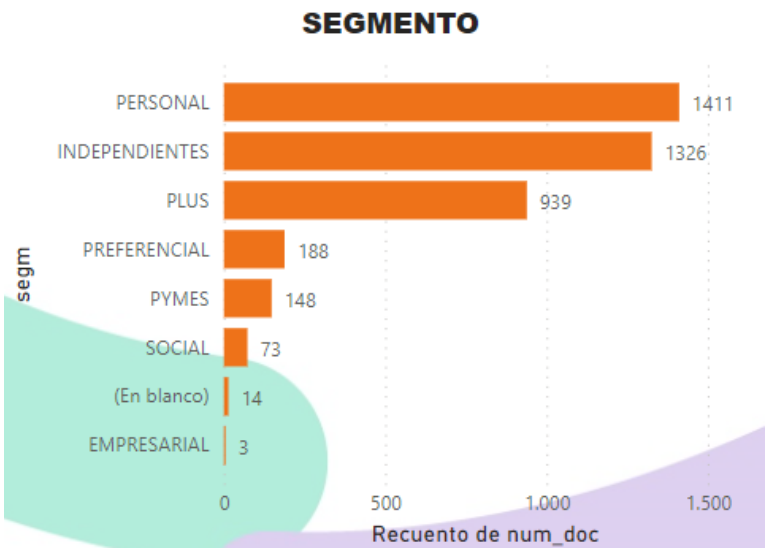
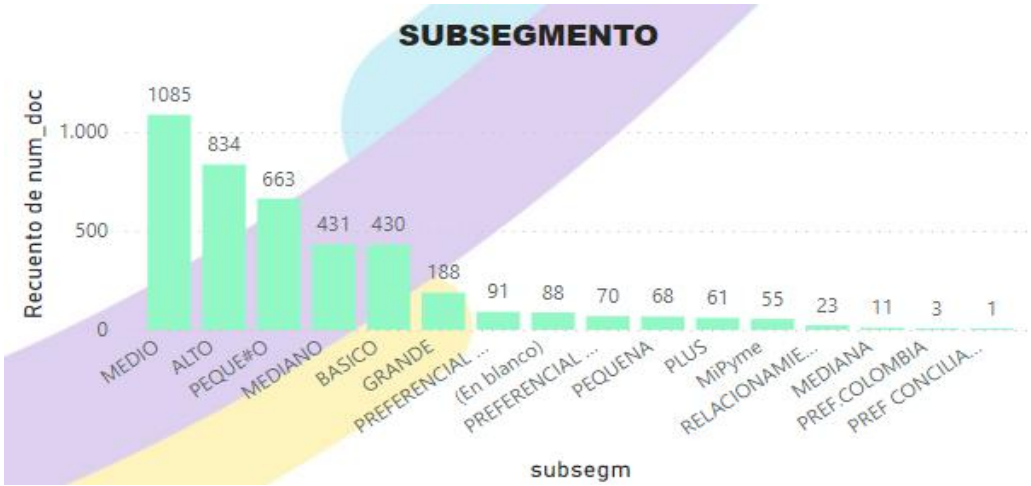


Figura 3

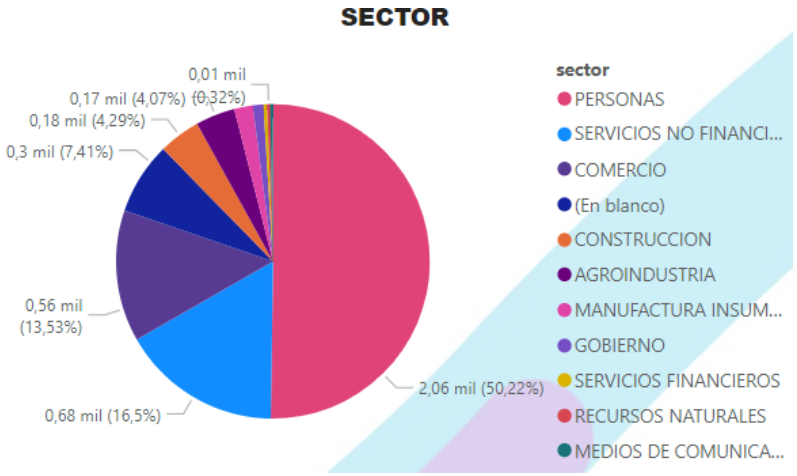
Subsegmentos de los clientes alertados por monitoreos transaccionales durante el primer semestre del 2023



- De acuerdo con la figura 4, el 50% de los clientes pertenecen al sector de personas, seguido por los sectores de servicios no financieros y comercio. Esto indica que, aunque los clientes presentan diferencias en términos de activos y transacciones, también comparten características homogéneas.

Figura 4

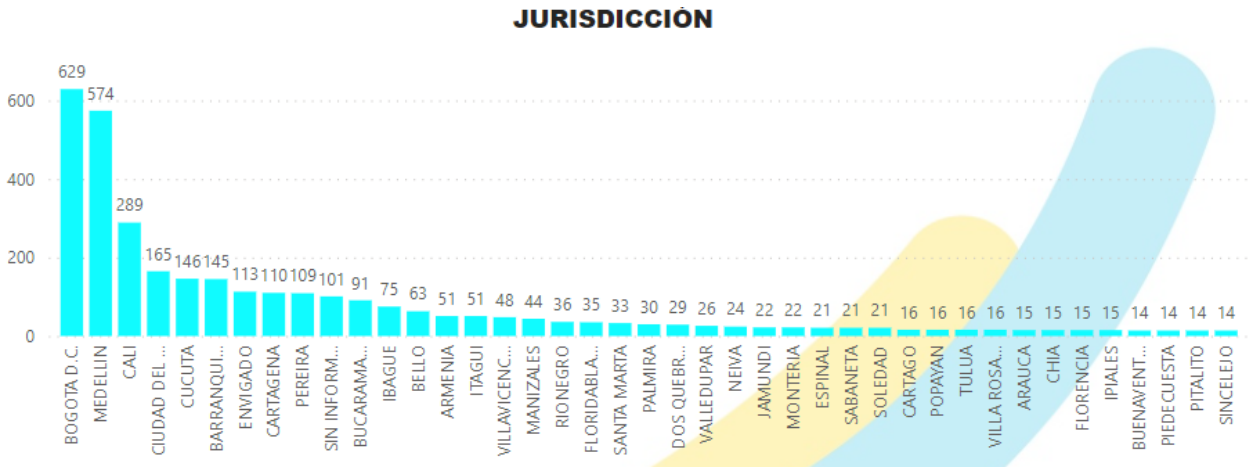
Sectores de los clientes alertados por monitoreos transaccionales durante el primer semestre del 2023



- La figura 5 muestra el mapeo realizado de los municipios donde residen los clientes alertados , ya que la jurisdicción es uno de los factores de riesgo LAFT. Se observa una concentración en ciudades principales como Bogotá y Medellín, lo cual es coherente debido a su alta población.

Figura 5

Jurisdicciones de los clientes alertados por monitoreos transaccionales durante el primer semestre del 2023



- Los clientes alertados tienen una concentración de productos de captación (cuentas de ahorro, cuentas corrientes, fondos de inversión colectiva) y productos de colocación (diferentes modalidades de crédito) lo cual se puede ver en la tabla 1.

Tabla 1

Tipo de productos de los clientes alertados por monitoreos transaccionales durante el primer semestre del 2023

TIPO PRODUCTO	FRECUENCIA	%
CAPTACIONES	12329	45.8%
COLOCACIONES	11099	41.3%
INVERSIONES	2915	10.8%
SERVICIOS	484	1.8%
UNIDAD INMOBILIARIA	66	0.2%

- En cuanto al comportamiento transaccional de entrada y salida de recursos a través de las cuentas de los clientes en cual está reflejado en la tabla 2, vimos que el 75% de los clientes han recibido menos de 70 millones y han realizado operaciones de salida con un valor total menor a 75 millones durante los seis meses analizados (primer semestre de 2023).

Tabla 2

Comportamiento transaccional de los clientes alertados por monitoreos transaccionales durante el primer semestre del 2023

	CREDITO	**DEBITO**
	monto	**monto**
count	31568	31302
mean	$ 206,226,228.59	$ 208,133,428.75
std	$ 2,635,905,575.72	$ 2,642,422,154.87
min	$ 0.01	$ 0.25
25%	$ 7,224,637.29	$ 8,967,236.77
50%	$ 22,983,082.37	$ 25,598,864.04
75%	$ 70,725,844.99	$ 75,825,418.10
max	$ 163,471,891,217.70	$ 155,616,187,866.45

- Se identificaron otras cualidades confidenciales derivadas de variables de manejo interno.

4.4. Preparación de los datos

Las actividades realizadas en esta etapa consisten en integrar los datos provenientes de diferentes fuentes (tablas) para preparar la matriz con las variables preliminares. Posteriormente, se llevó a cabo un análisis exploratorio para definir el dataset destinado al modelado. Primero, se aplicaron diversas transformaciones según el tipo de variable, se gestionó el tratamiento de datos faltantes y se establecieron los formatos adecuados para cada variable. Además, se crearon nuevas variables para evaluar su influencia en el riesgo LAFT basándose en la información disponible, y se realizaron agrupaciones para asegurar que cada cliente tuviera un solo registro mensual, incluso si fue alertado por múltiples monitoreos en el mismo mes.

Dado que la implementación del modelo será mensual, se seleccionaron las variables identificadas de las diferentes fuentes mencionadas en la etapa 2 y se llevaron a cabo las siguientes transformaciones:

- Encoder
- Variables binarias
- Variables ordinales
- Creación de nuevas variables y categorías

4.4.1. Variables categóricas a variables numéricas

Las variables "cliente sensible" y "cliente actualizado" se transformaron en variables binarias, tomando el valor 0 si no cumplían con la condición y el valor 1 si sí la cumplían. La variable "tipo de cliente", que indica si un cliente es persona natural o persona jurídica, se transformó asignando el valor 0 para persona natural y el valor 1 para persona jurídica.

4.4.2. Agrupación y creación de nuevas variables

- Se decidió agrupar la variable que especificaba el modelo analítico implementado según la clasificación de la Superintendencia Financiera, de acuerdo con la naturaleza de las operaciones a detectar. Esto permitió reducir las clases, ya que inicialmente se contaba con más de 15 modelos diferentes. Posteriormente, se aplicó un "encoding", resultando en 6 clases las cuales se pueden visualizar en la tabla 3.

Tabla 3

Monitoreos transaccionales por los cuales han sido alertados los clientes durante el primer semestre del 2023

Clasificación modelos	numero alertas
Transacciones_efectivo	2979
Operaciones_internacionales	1843
Operaciones_nacionales	458
Otras_transaccionales	27
campañas	27
Pep	9

- La variable que indica el segmento comercial al que pertenece cada cliente se trató como una variable categórica ordinal, reemplazando los segmentos por una escala numérica que depende del tamaño del cliente (ver tabla 4), considerando si era una persona natural (0) o una persona jurídica (1).

Tabla 4

Transformación de la variable segmento

tipo_cli	segm	jerarquia_segm
0	SOCIAL	1
0	PERSONAL	2
0	PLUS	3
0	PREFERENCIAL	4
0	INDEPENDIENTES	5
0	PYMES	6
1	PYMES	7
1	INTERNACIONAL	8
1	GOBIERNO	9
1	GOBIERNO DE RED	10
1	EMPRESARIAL	11

1	CONSTRUCTOR PYME	12
1	CONSTRUCTOR EMPRESARIAL	13
1	CONSTRUCTOR CORPORATIVO	14
1	CORPORATIVA	15
1	INSTITUCIONES FINANCIERAS	16

- Contábamos con una fuente que detallaba los productos pertenecientes al cliente, tanto activos como inactivos y cancelados. De esta fuente se tomó el campo "tipo de producto" para crear 6 nuevas variables que indican el número de productos que tiene un cliente por tipo de producto (ver tabla 5).

Tabla 5

Creación de la variable número de productos según el tipo de producto

VARIABLE	DESCRIPCIÓN
CAPTACIONES	número de productos de captación
SERVICIOS	número de productos de servicios
COLOCACIONES	número de productos de colocaciones
INVERSIONES	número de productos de inversión
UNIDAD INMOBILIARIA	número de productos de unidad inmobiliaria
NEGOCIOS FIDUCIARIOS	número de productos de negocios fiduciarios

- En la base de datos inicial que contenía las alertas y sus detalles, se utilizó la variable que indicaba el monto de cada alerta. Se sumó este valor por cliente y por mes de alerta, creando así otra variable llamada "cantidad de alertas" que contaba el número total de alertas agrupadas por cliente y mes. Esto resultó en un único registro por mes para cada cliente.
- Se aplicó un encoder a una de las variables que indican el manejo interno del control del riesgo LAFT, resultando en tres variables adicionales.
- Para las variables transaccionales, se agruparon por período mensual, distinguiendo entre entradas de recursos (MCA) y salidas de recursos (MDA). Se registró el monto total que ingresó y salió de las cuentas de cada cliente durante los diferentes meses del primer semestre de 2023.

También se crearon variables adicionales:

- Se contó la cantidad de alertas que un cliente había presentado antes del periodo alertado, así como cuántas de estas terminaron en un reporte de operación sospechosa, utilizando datos de dos fuentes diferentes.
- Se creó la variable "antigüedad" utilizando la fecha de vinculación del cliente, calculando así cuántos años ha sido cliente la persona.

4.4.3. Valores Faltantes

- La variable "ingresos", que indica el promedio de ingresos mensuales y es auto-declarada por el cliente, presentó la mayor cantidad de valores faltantes. Por lo tanto, se decidió imputar estos datos llenando los valores faltantes con el nivel de ingresos promedio del segmento al que pertenece el cliente como se puede ver en la tabla 6.

Tabla 6

Promedio de ingresos según el segmento

tipo_cli	segmento	Ingresos_promedio
0	SOCIAL	1,000,000
0	PERSONAL	2,000,000
0	PLUS	5,000,000
0	PREFERENCIAL	8,000,000
0	INDEPENDIENTES	41,000,000
0	PYMES	50,000,000
1	PYMES	50,000,000
1	INTERNACIONAL	50,000,000
1	GOBIERNO	50,000,000
1	GOBIERNO DE RED	50,000,000
1	EMPRESARIAL	4,200,000,000
1	CONSTRUCTOR PYME	10,000,000,000
1	CONSTRUCTOR EMPRESARIAL	10,000,000,000
1	CONSTRUCTOR CORPORATIVO	10,000,000,000
1	CORPORATIVA	10,000,000,000
1	INSTITUCIONES FINANCIERAS	10,000,000,000

- Se encontraron valores faltantes en MCA y MDA (entrada y salida de recursos) debido a que estos fueron agrupados por mes, y en algunos meses los clientes no realizaron movimientos entre sus cuentas, lo cual se verificó. Por lo tanto, se imputaron los datos llenando los valores faltantes con el número 0.

Además, algunas agrupaciones y filtros se realizaron durante la extracción en lenguaje SQL. Finalmente, se logró la construcción de una matriz inicial donde se consolidaron todas las variables, tanto las transformadas como las que no necesitaron ningún tratamiento. Esta matriz incluye información de las alertas del cliente, información demográfica, información transaccional, información sobre los productos gestionados en el banco y otra información relacionada con la mitigación del riesgo LAFT. Esto permitió realizar un análisis exploratorio y tomar decisiones con respecto a la construcción del dataset de entrenamiento.

4.4.4. Análisis exploratorio

En la primera parte del análisis exploratorio, se observó que no había datos faltantes en el dataset. Después de las transformaciones efectuadas sobre los datos, únicamente quedaron variables numéricas de tipo entero, flotante y binario. Luego, se generaron algunos estadísticos como desviación estándar, media, mínimo, máximo y cuartiles. A través de estas métricas, se pudo intuir una alta presencia de datos atípicos en las variables cantidad de alertas, alertas anteriores, cantidad de alertas anteriores que terminaron en un reporte, número de productos de captaciones, número de productos de colocaciones y número de productos de inversiones. En las variables que contienen información transaccional, la desviación estándar fue muy grande, por lo tanto, se concluyó que los datos estaban muy dispersos. También se corroboró que la media y la mediana de estos datos difieren mucho. Por estos hallazgos, se intuyó la presencia de datos atípicos en estas variables, además de la heterogeneidad de los datos. Durante el análisis, también se observaron algunos datos posiblemente erróneos en la edad y antigüedad de los clientes, pues se encontraron cuatro personas naturales con edades entre 1 y 17 años y seis personas naturales con una antigüedad superior a 112 años en el Banco.

Para un análisis más profundo, se separaron las variables en clases: numéricas, dummy y categóricas. En el análisis de los datos dummy, las variables categóricas resultantes de clasificar los modelos (otras transacciones, campañas, PEP) y otras variables como número de productos de negocios fiduciarios, marcación de si el cliente ha tenido una mención en medios de comunicación o en requerimientos por entes legales relacionados con actividades LAFT, marcación de si es una persona políticamente expuesta (PEP), marcación de si es una persona relacionada con PEP, si el cliente es importador, si es exportador, si maneja alto flujo de efectivo, si es una persona jurídica recién constituida, si es menor, si es un cliente sensible, categoría bloqueo, marcación de si es un empleado del banco, si ha sido reportado, la calificación de la jurisdicción, el tipo de cliente y las tres variables de manejo interno, tienen un desbalance muy grande porque entre el 95% y 99% de sus valores están en cero. Por medio de las variables dummy se pudo observar que el 56.02% de los clientes alertados durante el periodo analizado han realizado transacciones en efectivo. Realizar operaciones en efectivo por altos montos y de forma fraccionada es una señal de posible LAFT.

Para el análisis gráfico de las variables numéricas, se emplearon histogramas y Box-Plot. Por medio del Box-Plot, se pudo corroborar la presencia de datos atípicos, especialmente en las variables de monto alerta, ingresos mensuales, antigüedad, monto crédito, monto débito, alertas anteriores, colocaciones, recuento crédito y recuento débito. Los histogramas muestran que la variable edad presenta la forma más simétrica, por lo tanto, se intuye que los datos se distribuyen normalmente. En el resto de los histogramas se puede observar de forma muy clara una asimetría positiva, pues los datos están muy concentrados en la parte izquierda del histograma.

El análisis gráfico de las variables categóricas se hizo por medio de gráficos de barras. En las variables riesgo cliente y otros riesgos, hay un desbalance muy grande, pues sus datos convergen a un valor en particular. Por medio de la variable jerarquía segmento, se pudo observar que la mayoría de los clientes alertados correspondían a los segmentos Independientes y Personal, con el 34.24% y 31.89% de los clientes respectivamente. También se pudo validar que la mayoría de los clientes alertados tienen como calificación en el riesgo de la actividad económica, bajo y medio bajo, con el 58.94% y 37.86% de los clientes respectivamente. Esto quiere decir que las actividades económicas que registran en su mayoría los clientes con operaciones relacionadas con LAFT son de riesgo bajo para evitar otro tipo de sospechas. Lo mismo sucede con las variables riesgo cliente y RIC, que convergen a riesgos bajos para los clientes. Durante el periodo analizado, se pudo evidenciar que en marzo se generaron el 22.76% de las alertas, siendo este el mes con mayor cantidad de alertas.

Posteriormente, se realizó el análisis bivariado, inicialmente entre variables numéricas. Se empleó un gráfico de dispersión y se encontraron las siguientes relaciones lineales positivas entre las variables:

- Monto alertado e ingreso mensual: A mayor ingreso mensual, mayor es el monto alertado y viceversa.(figura 6)
- Monto alertado y monto crédito: A mayor monto crédito, mayor es el monto alertado y viceversa.
- Monto alertado y monto débito: A mayor monto débito, mayor es el monto alertado y viceversa.
- Edad y antigüedad: Entre más edad tiene el cliente, mayor es el número de años que lleva vinculado al banco.
- Ingreso mensual y monto crédito: A mayor ingreso mensual, mayor es el monto crédito

y viceversa.

- Ingreso mensual y monto débito: A mayor ingreso mensual, mayor es el monto débito y viceversa.
- Monto crédito y monto débito: A mayor monto crédito, mayor es el monto débito y viceversa. (figura 7)

Figura 6

Diagrama de dispersión del monto alertado vs. monto crédito acumulado.

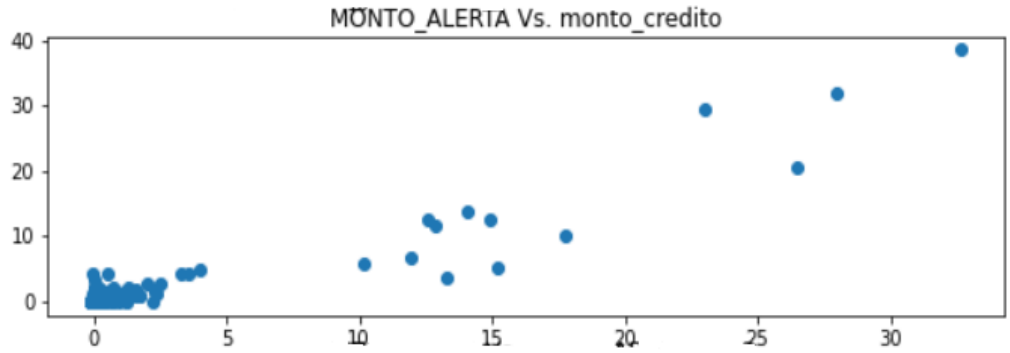
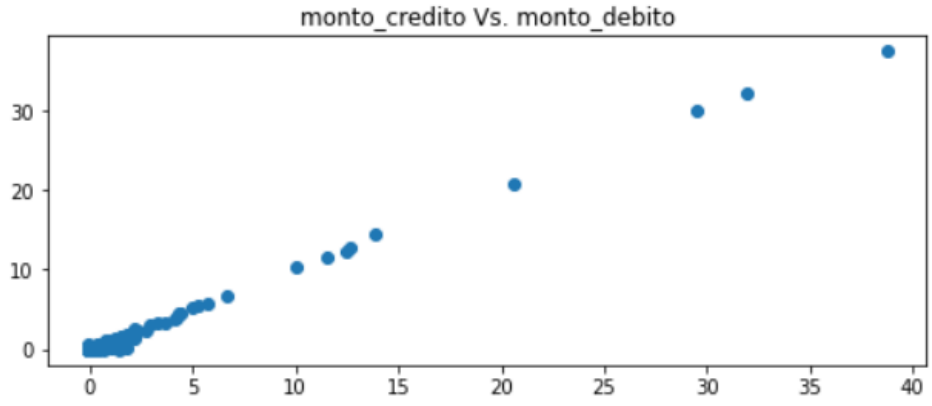


Figura 7

Diagrama de dispersión del monto crédito acumulado vs. monto débito acumulado.



Estos resultados muestran una relación entre variables que indican montos de dinero.

Luego del análisis bivariado efectuado entre las variables numéricas, se realizó un análisis entre las variables categóricas y se evidenció lo siguiente:

- Los clientes alertados de riesgo medio alto no tienen actividades económicas con calificaciones de alto riesgo. El 59.09% de estos clientes tienen actividades económicas de riesgo medio bajo, algo que no concuerda con su perfil.
- Todos los clientes alertados del segmento empresarial son de riesgo alto, con otros riesgos altos, y todas sus actividades económicas tienen riesgo medio.

- El 89.32% de los clientes alertados con actividad económica de riesgo alto tienen otros riesgos bajos, lo cual parece no ser coherente entre sí.

Después de esto, se hizo un análisis bivariado entre variables numéricas y categóricas:

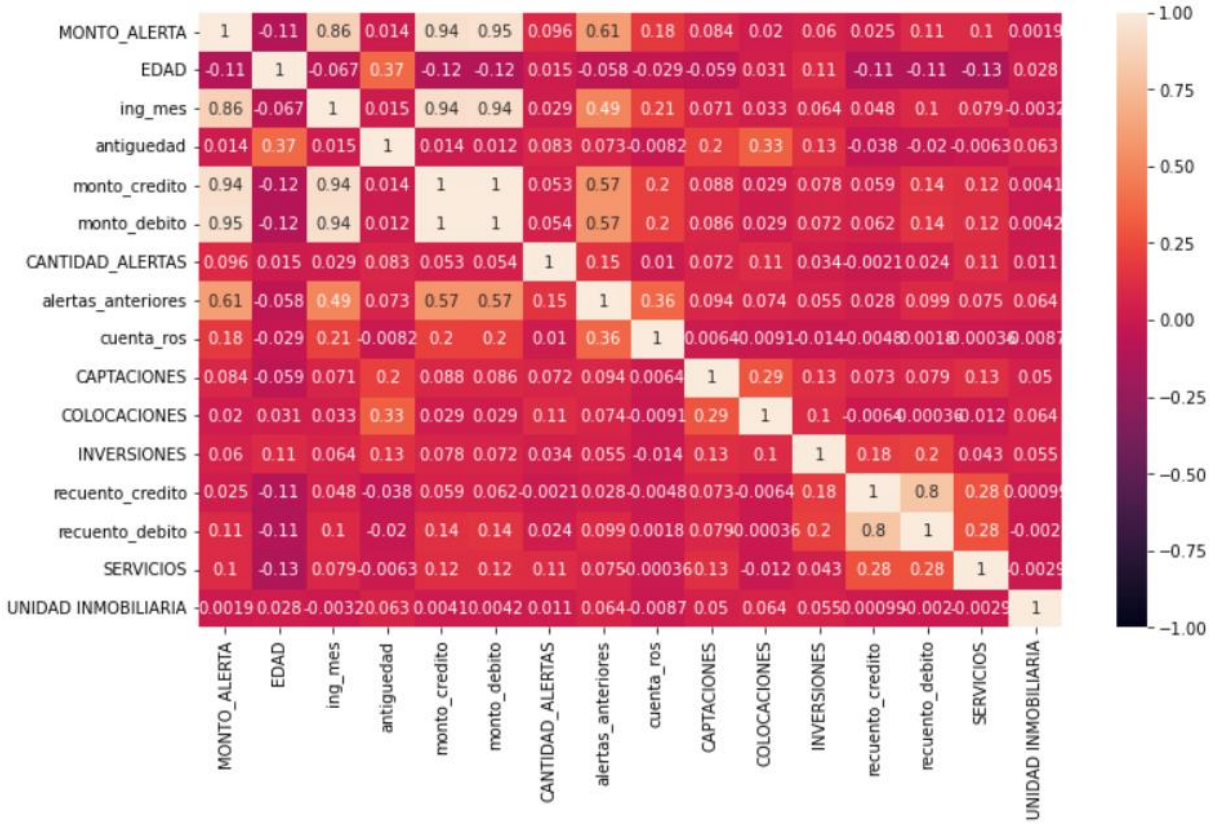
- El 35.24% y 32.96% del total del monto alertado pertenecen a los clientes con calificación baja y media en la actividad económica, respectivamente. Sin embargo, el promedio de monto alertado es mucho mayor en los clientes con calificación baja.
- Las personas naturales alertadas tienen en promedio una antigüedad entre 7 y 9 años en el Banco. De la misma forma, las personas jurídicas tienen una antigüedad entre 4 y 16 años. Por tanto, la mayoría de los clientes alertados no son recién vinculados o constituidos.
- Los clientes con la mayor cantidad de alertas son los de riesgo bajo en todas las categorías.
- El 45.67% y 41.97% de los clientes con calificación medio baja y baja en la calificación de la actividad económica, respectivamente, han tenido alertas anteriormente. Sin embargo, el promedio de alertas anteriores de los clientes con calificación medio baja en la actividad económica es de 8, mientras que ese mismo promedio es de 1 para los clientes con calificación baja en la actividad económica. Por tanto, los clientes con calificación medio baja reinciden más en operaciones sospechosas.

Por último, se realizó el análisis multivariado. Para las variables numéricas se empleó una matriz de correlaciones (ver figura 8) y se tomaron como significativas todas esas relaciones inferiores a -0.85 y superiores a 0.85. Se pudo observar lo siguiente:

- El coeficiente de correlación entre el ingreso mensual y el monto alertado es de 0.86, lo que significa que a mayor ingreso, mayor es el monto alertado y viceversa.
- El coeficiente de correlación entre el monto crédito y el monto alertado es de 0.94, lo que significa que a mayor monto crédito, mayor es el monto alertado y viceversa.
- El coeficiente de correlación entre el ingreso mensual y monto crédito es de 0.94, lo que significa que a mayor ingreso, mayor es el monto crédito y viceversa.
- El coeficiente de correlación entre el ingreso mensual y monto débito es de 0.94, lo que significa que a mayor ingreso, mayor es el monto débito y viceversa.
- El coeficiente de correlación entre el monto débito y el monto alertado es de 0.95, lo que significa que a mayor monto débito, mayor es el monto alertado y viceversa.
- El coeficiente de correlación entre monto crédito y monto débito es de 1, lo que significa que a mayor monto crédito, mayor es el monto débito y viceversa.

Figura 8

Matriz de correlación de las variables numéricas



Todos estos resultados respaldan lo evidenciado mediante los gráficos de dispersión.

Del análisis exploratorio efectuado se plantearon algunas modificaciones para poder emplear el dataset en el entrenamiento de los modelos:

- Se evidenció una gran cantidad de datos atípicos en muchas variables, lo cual puede afectar el entrenamiento de los modelos. Por tanto, se debe hacer el tratamiento de este tipo de datos.
- En las variables edad y antigüedad parecen haber datos erróneos. Estas variables deben explorarse más a fondo para saber a qué se deben estos datos.
- En las variables numéricas hay datos que difieren mucho entre sí, por lo que se puede considerar una estandarización de este tipo de variables.
- Muchas variables dummy no parecen aportar información relevante al dataset, por tanto, se va a hacer una selección de las variables dummy más importantes.
- Por medio del análisis bivariado y multivariado, se detectó una relación significativa entre

varias variables. Por tal razón, se van a eliminar las variables que están aportando información redundante.

Luego de abordar las cuatro etapas expuestas anteriormente, dimos cumplimiento al primer objetivo específico, y como resultado obtuvimos un dataset con 4939 registros y 30 atributos que dan información de la persona alertada, la periodicidad de la alerta, el origen de esta y las características demográficas y de riesgo de la persona, lo cual comprende la mayoría de las variables que son tenidas en cuenta por los investigadores a la hora de analizar un caso de investigación.

Ya que se contaba con pocos registros debido a que el planteamiento del proyecto se llevó a cabo en el primer semestre del 2023 y solo se tomarían los clientes alertados en dicho periodo, se tomó la decisión de agregar más datos ya que había la posibilidad de hacerlo. Dado que se culminó todo el 2023 y se contaba con toda esta data para ser procesada y así ampliar nuestro set de datos, lo cual beneficiaría directamente el rendimiento del modelado. Por lo tanto, el resultado final es un set de datos con 12204 registros y 30 clases o atributos.

5. ENTRENAMIENTO DE MODELOS DE CLUSTERIZACIÓN CON DIFERENTES TÉCNICAS DE APRENDIZAJE NO SUPERVISADO.

Durante la etapa de modelado, se realizaron varias tareas previas a la construcción de los modelos. En primer lugar, se exploraron diversas técnicas de aprendizaje no supervisado de clustering utilizando la librería Pycaret. Se entrenaron los modelos disponibles de forma básica, empleando los hiperparámetros por defecto y aplicando normalización a los datos. El objetivo de este enfoque inicial fue obtener un primer acercamiento a las métricas generadas por las diferentes técnicas las cuales se pueden ver en la tabla 7, para así decidir en cuáles enfocarse con mayor certeza respecto a los posibles resultados. En lugar de ajustar todos los hiperparámetros para los distintos modelos, se utilizó el método del codo (Elbow) para determinar el valor óptimo de K (número de clústeres) en las técnicas KMeans, HClust y Birch. Estas técnicas requieren definir adecuadamente el hiperparámetro K, ya que de este depende en gran medida el resultado del modelado.

Tabla 7

Técnicas de aprendizaje no supervisado empleadas

Name	Reference
K-Means Clustering	sklearn.cluster._kmeans.KMeans
Affinity Propagation	sklearn.cluster._affinity_propagation.AffinityPropagation
Mean Shift Clustering	sklearn.cluster._mean_shift.MeanShift
Spectral Clustering	sklearn.cluster._spectral.SpectralClustering
Agglomerative Clustering	sklearn.cluster._agglomerative.AgglomerativeClustering
Density-Based Spatial Clustering	sklearn.cluster._dbscan.DBSCAN
OPTICS Clustering	sklearn.cluster._optics.OPTICS
Birch Clustering	sklearn.cluster._birch.Birch

Los resultados del entrenamiento de los modelos fueron los siguientes:

Tabla 8

Métricas de calidad del entrenamiento de los modelos

Modelo	Silhouette	Calinski-Harabasz	Davies-Bouldin	Homogeneity	Rand Index	Completeness
meanshift	0.27	98.41	0.83	0	0	0
birch	0.17	794.24	1.2	0	0	0
ap	0.17	182.54	1.29	0	0	0
hclust	0.17	811.16	1.41	0	0	0
kmeans	0.15	852.14	1.35	0	0	0
optics	-0.35	13.73	1.34	0	0	0
dbscan	-0.45	3.96	1.6	0	0	0

De acuerdo con la tabla 8, se pudo evidenciar que los resultados del entrenamiento de los modelos no fueron satisfactorios, lo cual era previsible dado que no se realizó una sintonización de sus hiperparámetros. Por lo tanto, se seleccionaron algunas técnicas específicas, basándose en las recomendaciones de expertos en el área y en los resultados obtenidos por ellos en tareas similares. Posteriormente, se llevó a cabo la construcción de los modelos de manera manual, siguiendo las premisas particulares de cada técnica.

5.1. Aplicación de Kmeans

La primera técnica seleccionada fue K-means, debido a que es un modelo relativamente sencillo, muy poderoso y utilizado en este tipo de problemas. Se tuvieron en cuenta las siguientes premisas:

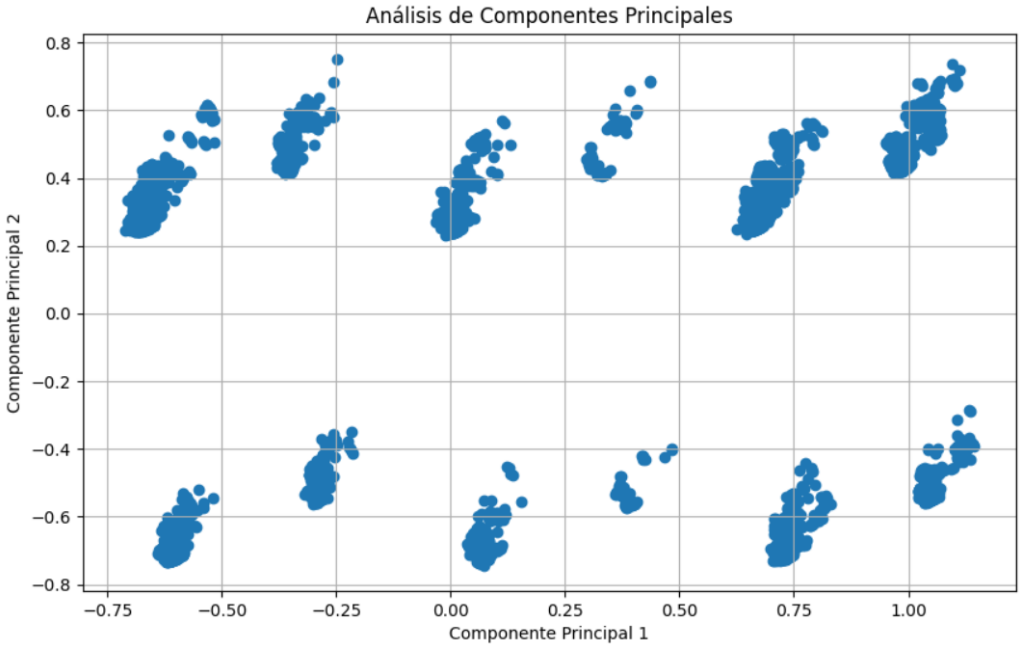
- Para predecir el grupo de un nuevo punto, determina cuál es el centroide más cercano (calcula la distancia).
- Es un modelo que se basa en distancia; por lo tanto, es recomendable utilizar los datos normalizados.
- Es apropiado para datos cuya estructura es circular o esférica. Para validar si una base de datos tiene una estructura circular o esférica, se utilizan técnicas de análisis multivariante, como el análisis de componentes principales (PCA) o pruebas estadísticas específicas para evaluar la esfericidad de los datos, como la prueba de esfericidad de Bartlett o la prueba de esfericidad de Mendoza.

  PCA: Técnica de reducción de dimensionalidad que ayuda a visualizar la estructura de los datos en un espacio de menor dimensión. Ayuda a identificar patrones y relaciones ocultas en los datos originales. Si los datos se distribuyen de manera circular o esférica en este espacio, esto puede indicar que la estructura de los datos en la base de datos original también tiene esa forma. Esta información es crucial para entender la naturaleza intrínseca de los datos y para seleccionar adecuadamente los métodos de modelado y análisis posteriores.

```
# Aplica PCA
pca = PCA(n_components=2)  # Reducir a 2 dimensiones para visualización
principal_components = pca.fit_transform(scaled_data)
```

Figura 9

Gráfico de dos dimensiones del PCA



El PCA se aplicó para reducir la dimensionalidad a 2 componentes principales (n_components=2) para poder visualizar los datos en un gráfico de dispersión (figura 9). Al visualizar los datos en el espacio de los componentes principales, podemos observar que estos podrían tener una estructura circular o esférica en pequeños grupos. Aunque la varianza explicada por las dos componentes es solo del 54%, nos da una idea de la estructura de los datos originales en un espacio de mayor dimensión.

Prueba de esfericidad de Bartlett: Se evaluó la hipótesis nula de que las variables son estadísticamente independientes, lo cual sugiere que la matriz de correlación es una matriz de identidad. Esto implicaría que los datos tienen una estructura esférica. Para esto, se comparó una matriz de correlación (utilizando correlaciones de Pearson) con la matriz de identidad. En resumen, se verificó si existía redundancia entre variables que pudiera ser resumida con algunos factores.

```
# Calcula la prueba de esfericidad de Bartlett
chi2_statistic = -np.log(det_correlation_matrix) * (n_variables - 1 - (2*n_variables + 5) / 6)
degrees_of_freedom = (n_variables*(n_variables-1)) / 2
p_value_bartlett = 1 - stats.chi2.cdf(chi2_statistic, degrees_of_freedom)

Resultado:

Estadístico de Chi-cuadrado: 101.68255070877967
Valor p: 1.0
No se rechaza la hipótesis nula: las variables podrían tener estructura esférica.
```

Como el valor p es mayor al nivel de significancia elegido (0.05), se puede concluir que las variables tienen una estructura esférica y, por lo tanto, no están correlacionadas de manera significativa. Esto significa que las variables son independientes entre sí y que la

matriz de correlación es una matriz de identidad.

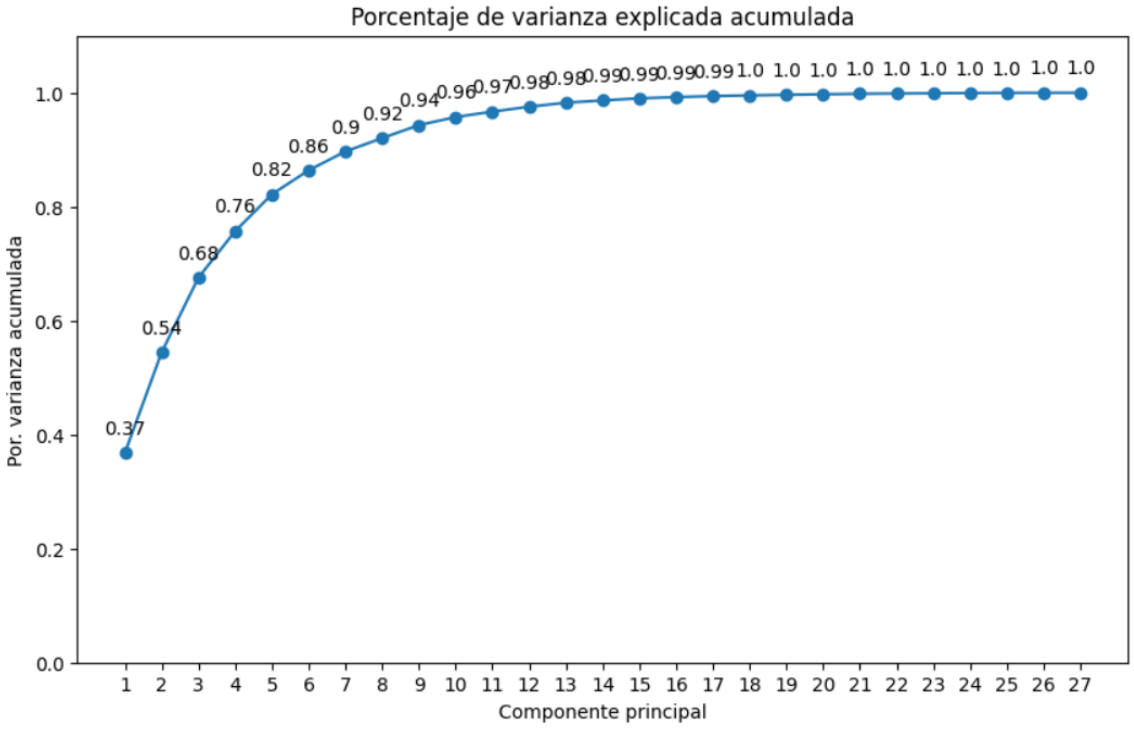
Después de realizar estos pasos, llegamos a la conclusión de que la base de datos podría ser adecuada para implementar el modelo de KMeans.

Antes de implementar esta técnica, se realizó un análisis para entender cuánta varianza de los datos se explica al aplicar reducción de dimensionalidad con diferentes cantidades de componentes. Se evaluó desde 1 componente hasta un máximo de 27, que es la dimensionalidad original del dataset. Esto permitio visualizar en la figura 10 cómo la cantidad de componentes afecta la capacidad de explicar la variabilidad de los datos.

```
x5 = scaled_data.copy()
pca =  PCA(n_components = 27)
pca.fit(x5)
```

Figura 10

Varianza explicada acumulada del PCA



Se seleccionó el número de componentes basándose en el porcentaje de varianza que se deseaba retener, en este caso el 96%. Esto se logró con 10 componentes principales. Se utilizó este conjunto reducido de componentes para entrenar el modelo, y posteriormente se comparó con el modelo entrenado utilizando las variables originales.

```
# Aplicar PCA con 10 componentes
pca =  PCA(n_components = 10)
new_principal_components = pca.fit_transform(x5)
```

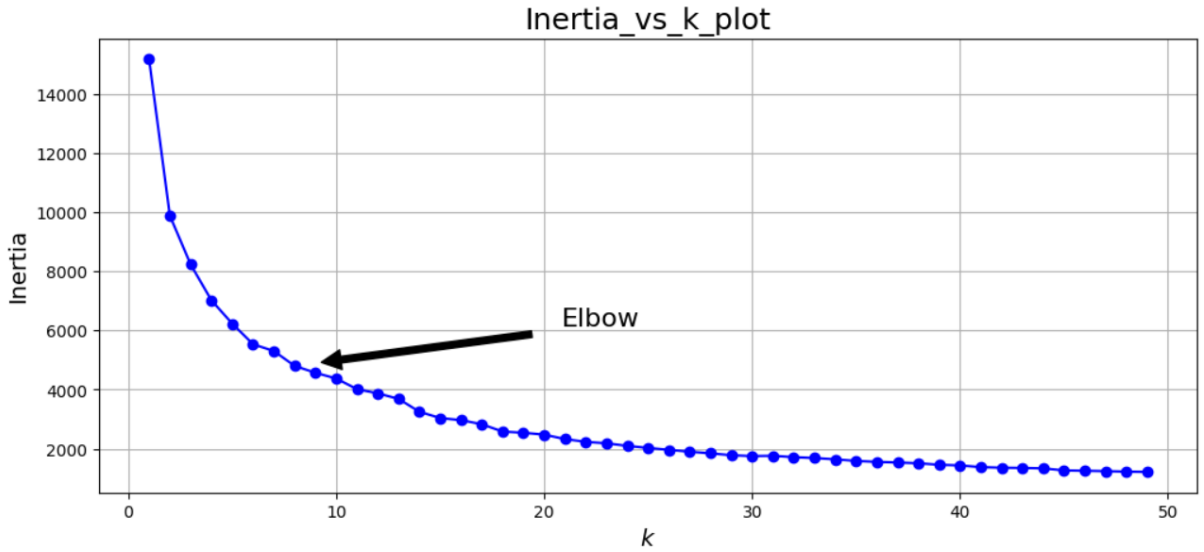
5.1.1. Modelo 1: Kmeans con PCA=10

K-means utiliza una métrica de rendimiento llamada inercia del modelo, que representa la suma de las distancias cuadráticas entre cada muestra y su centroide más cercano. El objetivo es minimizar esta inercia para obtener clústeres más compactos y definidos.

- **Paso 1:** Se procedió a realizar el gráfico del codo para determinar el número óptimo de clústeres (k) para nuestro conjunto de datos (figura 11). Se ajustaron varios modelos K-means con diferentes valores de k y se calculó la inercia para cada uno. Los resultados obtenidos fueron los siguientes:

Figura 11

Gráfico del codo para K-Means con PCA = 10



En la figura 11, se observó que la inercia disminuyó rápidamente al principio a medida que se aumentó k, pero luego disminuyó más lentamente. Se identificó un posible codo en k=8; sin embargo, también se consideraron otros valores como k=15, 20 y 25 debido a que mostraron cambios en la gráfica y son números de clústeres más apropiados para el tema en cuestión. Para tomar una decisión más informada sobre el número óptimo de clústeres, se utilizaron otras métricas que evaluaron la cohesión y separación de los clústeres en los siguientes pasos. La inercia no es una métrica ideal para determinar el número óptimo de clústeres (k) en KMeans, ya que tiende a disminuir a medida que aumenta k, dado que cada instancia tiende a estar más cerca de su centroide más cercano conforme se agregan más clústeres.

- **Paso 2:** Se implementó el coeficiente de silueta como complemento al método del codo para determinar el número óptimo de clústeres (ver figura 12). El coeficiente de silueta es una medida que considera tanto la cohesión dentro de un clúster como la separación entre clústeres. Esto permite una evaluación más precisa de la calidad de los clústeres en comparación con el método del codo.

  Donde:

- a es la distancia promedio de cada muestra con respecto a todas las instancias del mismo grupo.
- b es la distancia promedio entre cada muestra y las muestras del grupo más cercano (excluyendo el grupo al que pertenece la muestra en análisis).

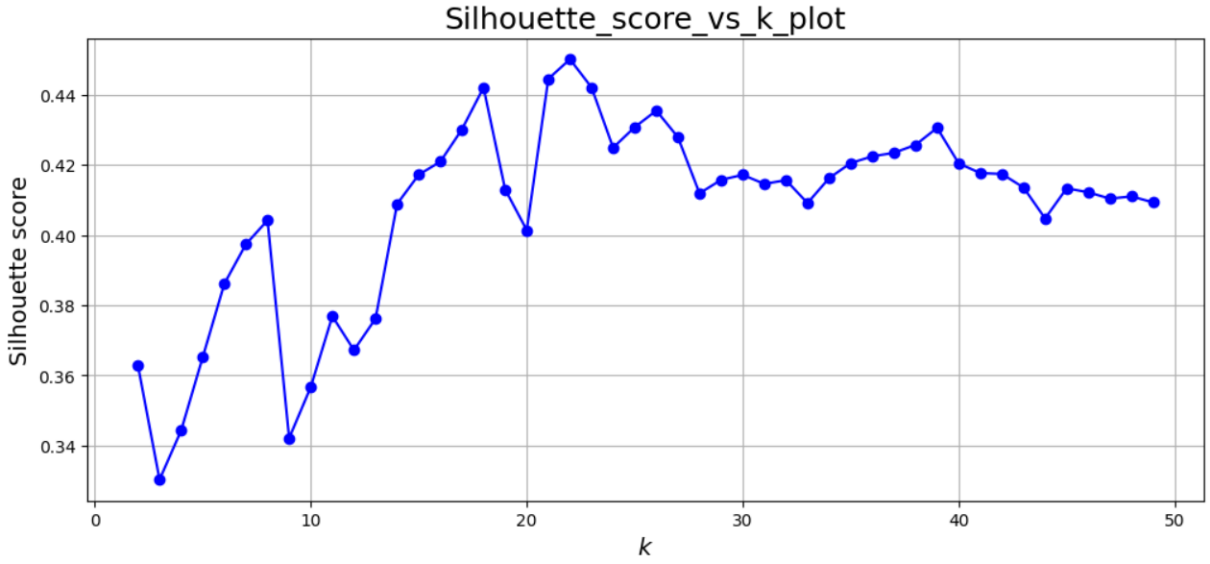
$$\frac{b-a}{\max(a,b)}$$

El coeficiente de silueta puede tomar valores entre -1 y 1. Un coeficiente cercano a 1 indica que la instancia está cerca de su centroide y lejos de otros grupos, lo que sugiere una buena separación. Un coeficiente cercano a 0 indica que la muestra está ubicada cerca de la frontera de un grupo. Un coeficiente negativo indica que la muestra está más cercana a otro grupo que al que fue asignada inicialmente, lo cual sugiere una asignación incorrecta del clúster.

```
silhouette_scores = [silhouette_score(new_principal_components, model.labels_)
                     for model in kmeans_per_k[1:]]
```

Figura 12

Gráfico del codo para K-Means con PCA = 10



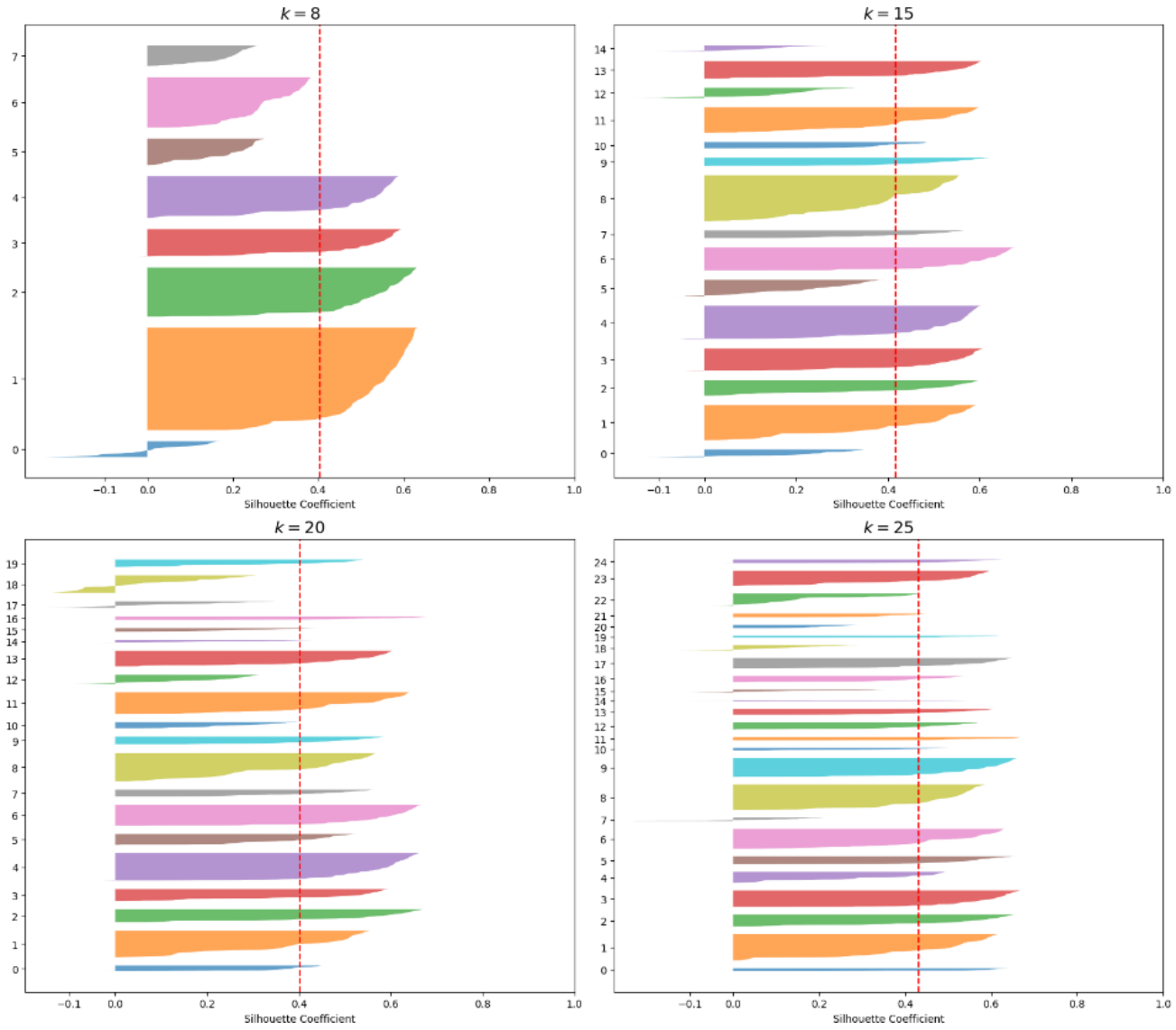
Al observar el gráfico del coeficiente de silueta, se notó que los valores estaban más cercanos a 0 que a 1 para cualquier k. Específicamente, se observó que cuando k=8, el coeficiente presentaba una métrica mejor que cuando k=10. Sin embargo, al examinar el resto del gráfico, se encontraron resultados superiores en k=20 y k=23. A pesar de esto, ninguno de estos valores se acercaba a 1, que era el valor deseado.

- **Paso 3:** Dado que en ocasiones quedarse con el valor de k que maximice el coeficiente de silueta no es la mejor opción, se pueden tomar decisiones más informadas al utilizar el diagrama de silueta (figura 13). Este diagrama grafica el coeficiente de silueta de cada una de las muestras, organizadas por cada grupo y en

orden descendente según el valor del coeficiente. En este caso, se graficaron los valores para k=8, 15, 20 y 25.

Figura 13

Diagrama de silueta para K-Means con PCA = 10



La altura de cada figura en forma de cuchillo indica la cantidad de muestras en cada grupo, el ancho de cada figura indica el valor de silueta de cada muestra y la línea vertical punteada indica el valor de silueta promedio de cada grupo. Se buscan grupos homogéneos en número de muestras y que estén encima del promedio.

Teniendo en cuenta los tres pasos anteriores, se tomó la decisión de implementar k=25, ya que este valor muestra las mejores métricas según los pasos anteriores y parece más adecuado para los resultados esperados. Sin embargo, se observa que

para todos los valores de k el promedio del coeficiente de silueta es muy bajo, lo que sugiere que los grupos formados pueden no ser de alta calidad.

Se entrena el modelo con las 10 componentes principales y con un número de clúster igual a 25.

```
# Entrenar el modelo
kmeans1 = KMeans(n_clusters=25)
kmeans1.fit(new_principal_components)

# Predicciones de los clusters
y_kmeans1 = kmeans1.predict(new_principal_components)
```

Las métricas resultantes son:

```
Silhouette Score: 0.42993310240018484
Calinski Harabasz Score: 3261.856712745914
Davies Bouldin Score: 1.0113998394380597
Inertia: 2042.1508829212569
```

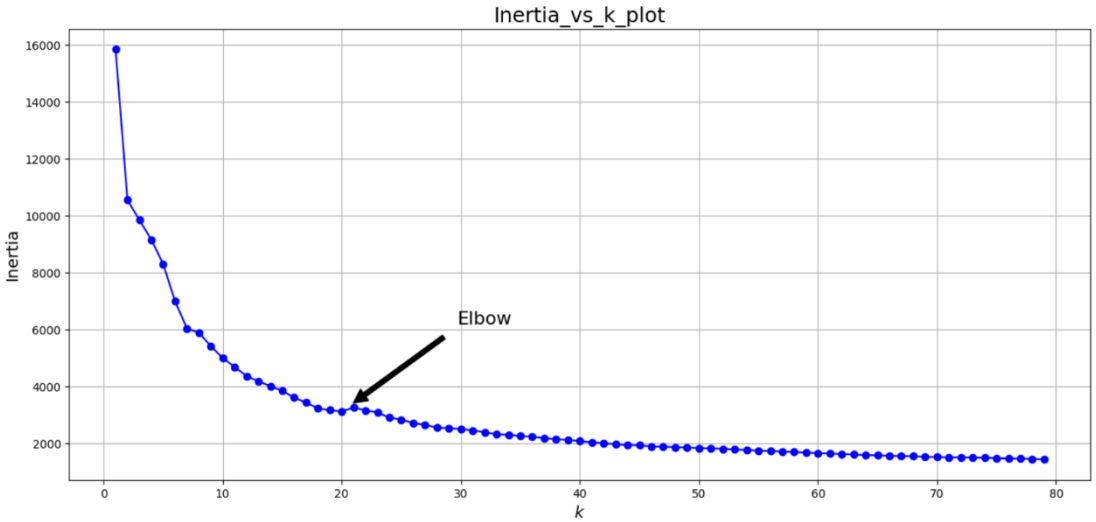
5.1.2. Modelo 2: Kmeans con el set de datos normalizados

Como se mencionó anteriormente, se aplica la técnica de k-means también al conjunto de datos originales normalizados, siguiendo los mismos tres pasos anteriores.

- **Paso 1:** Análisis de la inercia para varios valores de k. De acuerdo con el grafico en la figura 14, no es muy claro cuál sería el punto de inflexión ("elbow"). Por lo tanto, se tomaron en cuenta los valores de k=10, 20, 30 y 40 para los siguientes pasos.

Figura 14

Gráfico del codo para K-Means con el set de datos normalizados

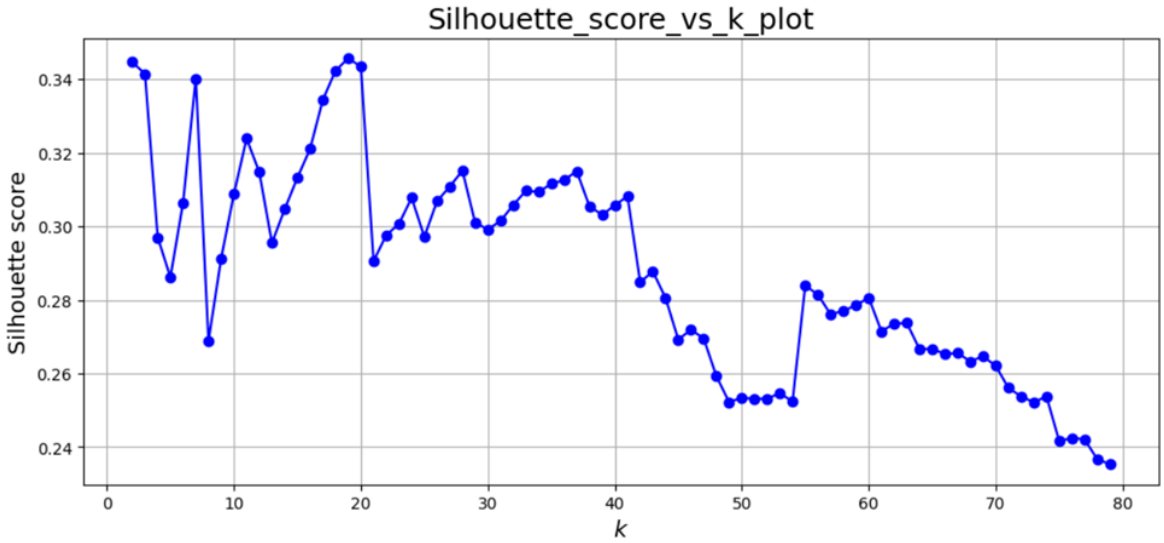


- **Paso 2:** Análisis del score de silueta para varios valores de k. Al observar el gráfico del coeficiente de silueta (figura 15), notamos que los valores están más cercanos a 0 que a 1 para cualquier valor de k. Es importante destacar que los coeficientes para

k=19 y k=20 muestran métricas superiores a todos los demás valores de k, incluidos 10, 30 y 40, que originalmente se consideraron.

Figura 15

Gráfico del coeficiente de silueta para K-Means con el set de datos normalizados

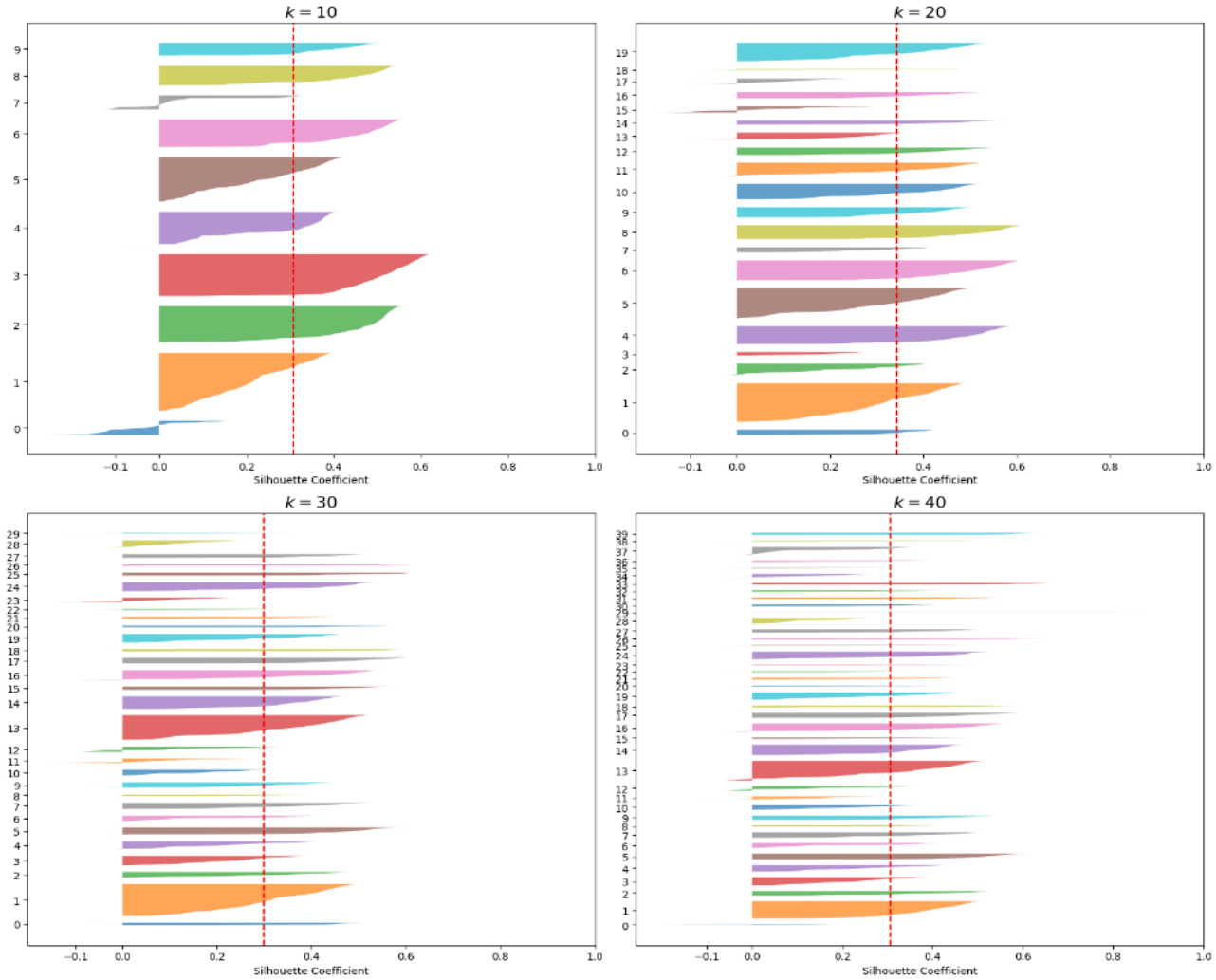


- **Paso 3:** Diagrama de silueta.
  Observando los diagramas de silueta con k=10, 20, 30 y 40 en la figura 16, observamos que no se forman clusters homogéneos en cuanto al número de muestras. Además, se observan datos mal clasificados y promedios de coeficiente de silueta muy bajos, donde muchas muestras no superan este valor. Esto es evidente en el diagrama donde k=10, que muestra el promedio más alto del coeficiente de silueta. Sin embargo, este k no es óptimo para la solución del problema planteado, por lo que se descarta el uso del modelo k-means con el conjunto total de datos originales normalizados. En su lugar, se procede a entrenar el modelo para almacenar las métricas y realizar comparaciones finales.

```
# Entrenar el modelo
kmeans2 = KMeans(n_clusters=10)
kmeans2.fit(scaled_data)

# Predicciones de los clusters
y_kmeans2 = kmeans2.predict(scaled_data)
```

Figura 16

Diagrama de silueta para K-Means con el set de datos normalizados



las métricas resultantes son:

Silhouette Score: 0.3225778765850447
Calinski Harabasz Score: 3198.757559514002
Davies Bouldin Score: 1.2143505610391334
Inertia: 4714.469272220894

5.1.3. Modelo 3: Kmeans con PCA=5

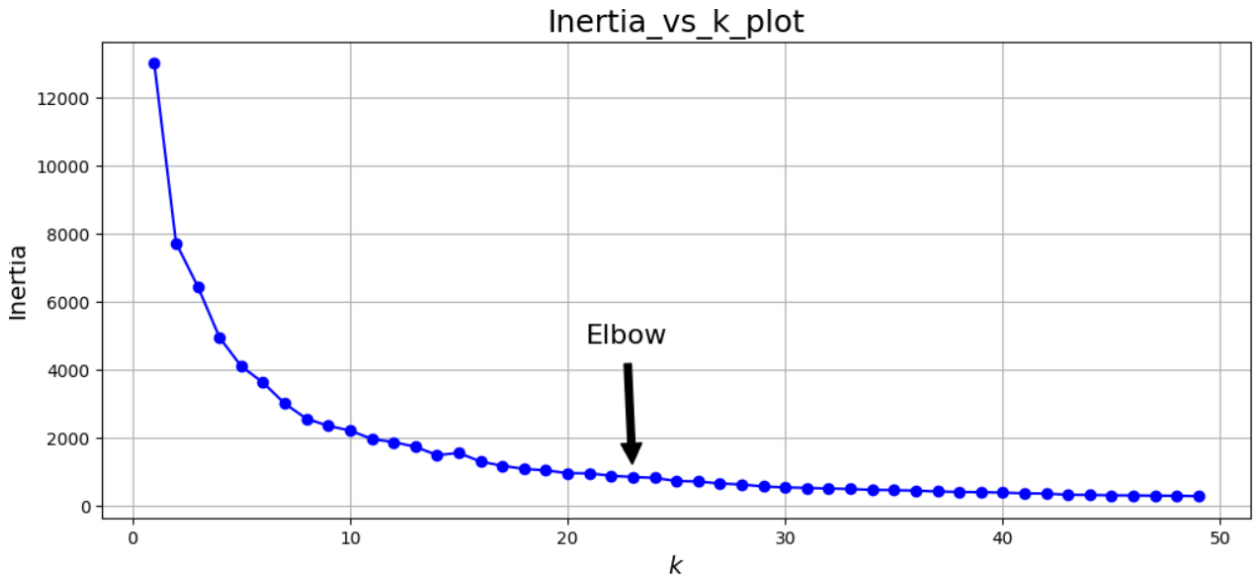
Se aplicó la técnica de KMeans a las nuevas 5 componentes resultantes después de aplicar PCA, las cuales explican el 82% de la variabilidad de los datos originales. Esta estrategia se utilizó para mejorar el rendimiento del algoritmo, ya que puede verse afectado por la maldición de la dimensionalidad en conjuntos de datos de alta dimensionalidad. La maldición de la dimensionalidad hace que las distancias entre puntos sean menos significativas a medida que aumenta el número de dimensiones, lo que puede resultar en agrupamientos menos precisos.

```
# Aplica PCA con 5 componentes
pca = PCA(n_components=5)  # Reducir a 2 dimensiones para visualización
principal_components5 = pca.fit_transform(scaled_data)
```

- **Paso 1:** análisis de la inercia para varios valores de k. De acuerdo con el gráfico (figura 17), no es muy claro cuál sería el punto de inflexión ("elbow"). Por lo tanto, se procedió a realizar el paso 2 para seleccionar los valores de k que se tuvieron en cuenta en el diagrama de silueta.

Figura 17

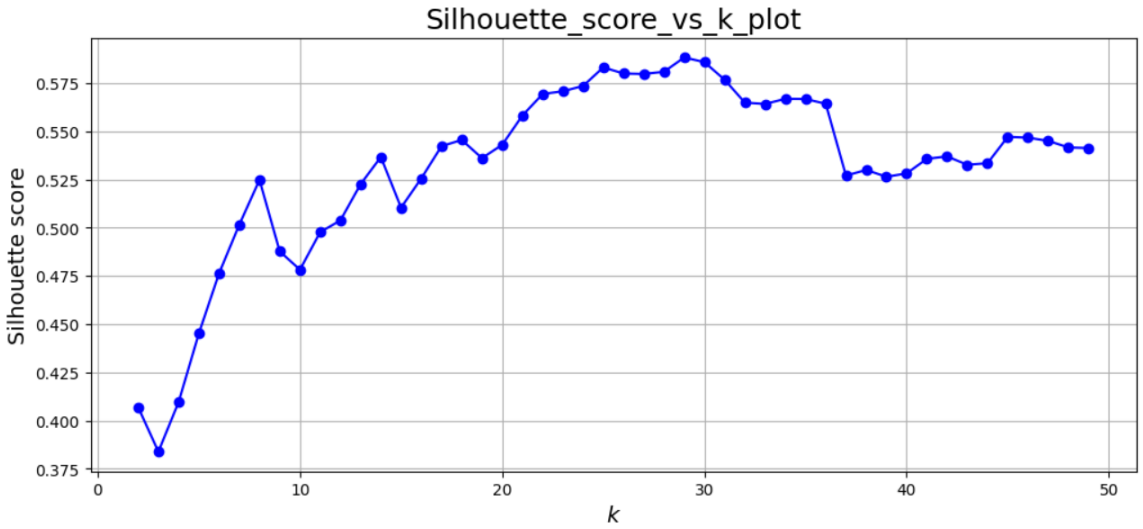
Gráfico del codo para K-Means con PCA = 5



- **Paso 2:** análisis del score de silueta para varios valores de k. Después de analizar la gráfica del score de silueta vs el número de k (figura 18), pudimos observar que los valores de k con mejores puntuaciones son k=8, 25 y 30. También se tomó en cuenta k=40, ya que este número de clústeres se consideró más beneficioso para la solución del problema.

Figura 18

Gráfico del coeficiente de silueta para K-Means con PCA = 5



- **Paso 3:** Diagrama de silueta (figura 19). Al analizar el diagrama de silueta para los diferentes valores de k seleccionados, observamos que solo hay muestras mal clasificadas en k=8, por lo que fue descartado debido a que no se ajusta a la solución esperada. En cambio, se seleccionó k=40, ya que la mayoría de las muestras tienen un score de silueta por encima del promedio y se observan grupos más homogéneos en cuanto al número de muestras, aunque algunos grupos presentan mayor concentración.

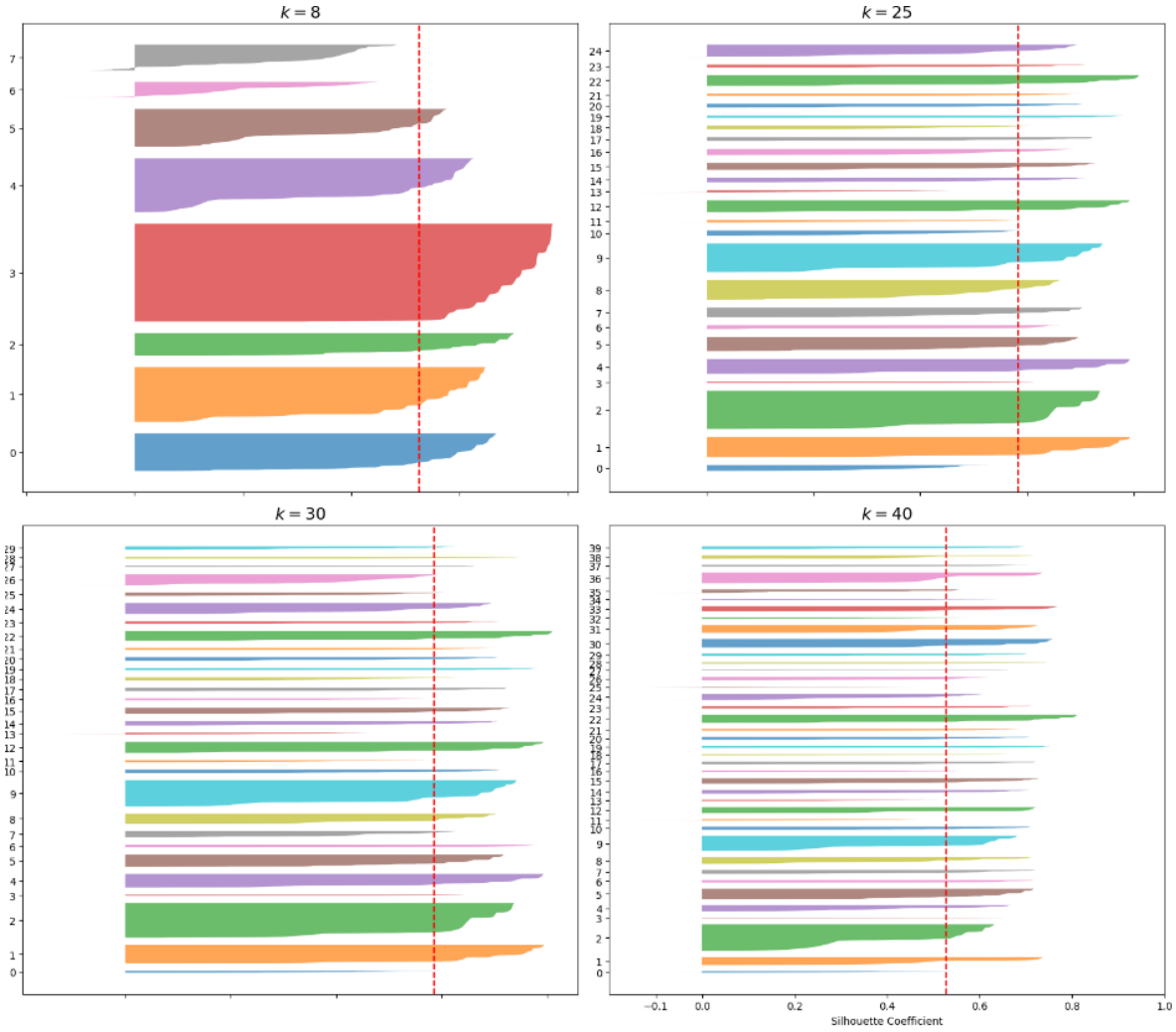
```
# Entrenar el modelo
kmeans3 = KMeans(n_clusters=40)
kmeans3.fit(principal_components5)

# Predicciones de los clusters
y_kmeans3 = kmeans3.predict(principal_components5)
```

las métricas resultantes son:

```
Silhouette Score: 0.5353523971855146
Calinski Harabasz Score: 10761.823669765803
Davies Bouldin Score: 0.7076876257002215
Inertia: 366.7447547493653
```

Figura 19

Diagrama de silueta para para K-Means con PCA = 5



5.1.4. Modelo 4: Kmeans con PCA=15

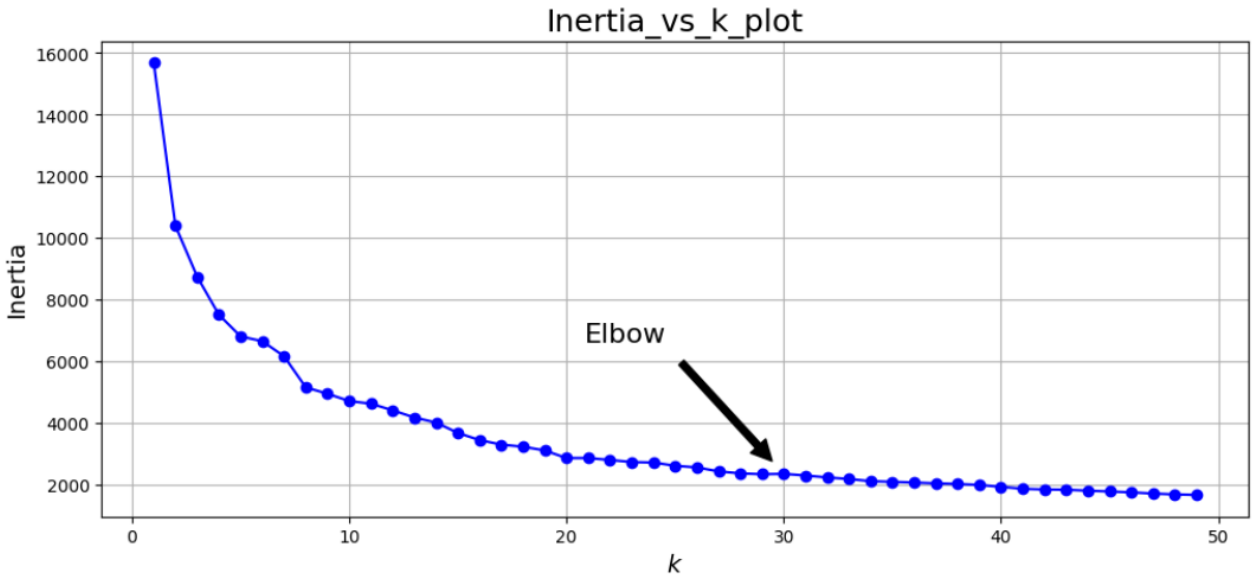
Por último, se aplicó la técnica de k-means con 15 componentes, ya que estas explican el 99% de la variabilidad de los datos, reduciendo la dimensión de 27 a 15.

```
# Aplicar PCA con 15 componentes
pca =  PCA(n_components = 15)
principal_components15 = pca.fit_transform(x5)
```

- **Paso 1:** análisis de la inercia para varios valores de k. De acuerdo con el gráfico (ver figura 20), no es muy claro cuál sería el punto de inflexión ("elbow"). Por lo tanto, se procederá a realizar el Paso 2 para seleccionar los valores de k que se tomarán en cuenta para el diagrama de silueta.

Figura 20

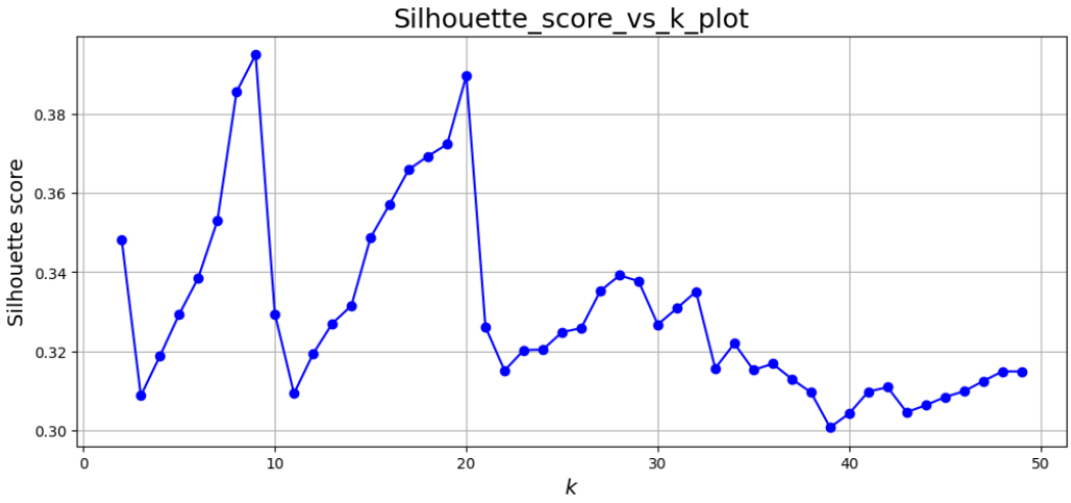
Gráfico del codo para K-Means con PCA = 15



- **Paso 2:** análisis del score de silueta para varios valores de k. Se analizó la gráfica del score de silueta vs el número de k (figura 21), donde se observó que los valores de k con mejores puntuaciones fueron k=9, 19, 20 y 32. Sin embargo, estos valores también mostraron scores de silueta más cercanos a 0 que a 1.

Figura 21

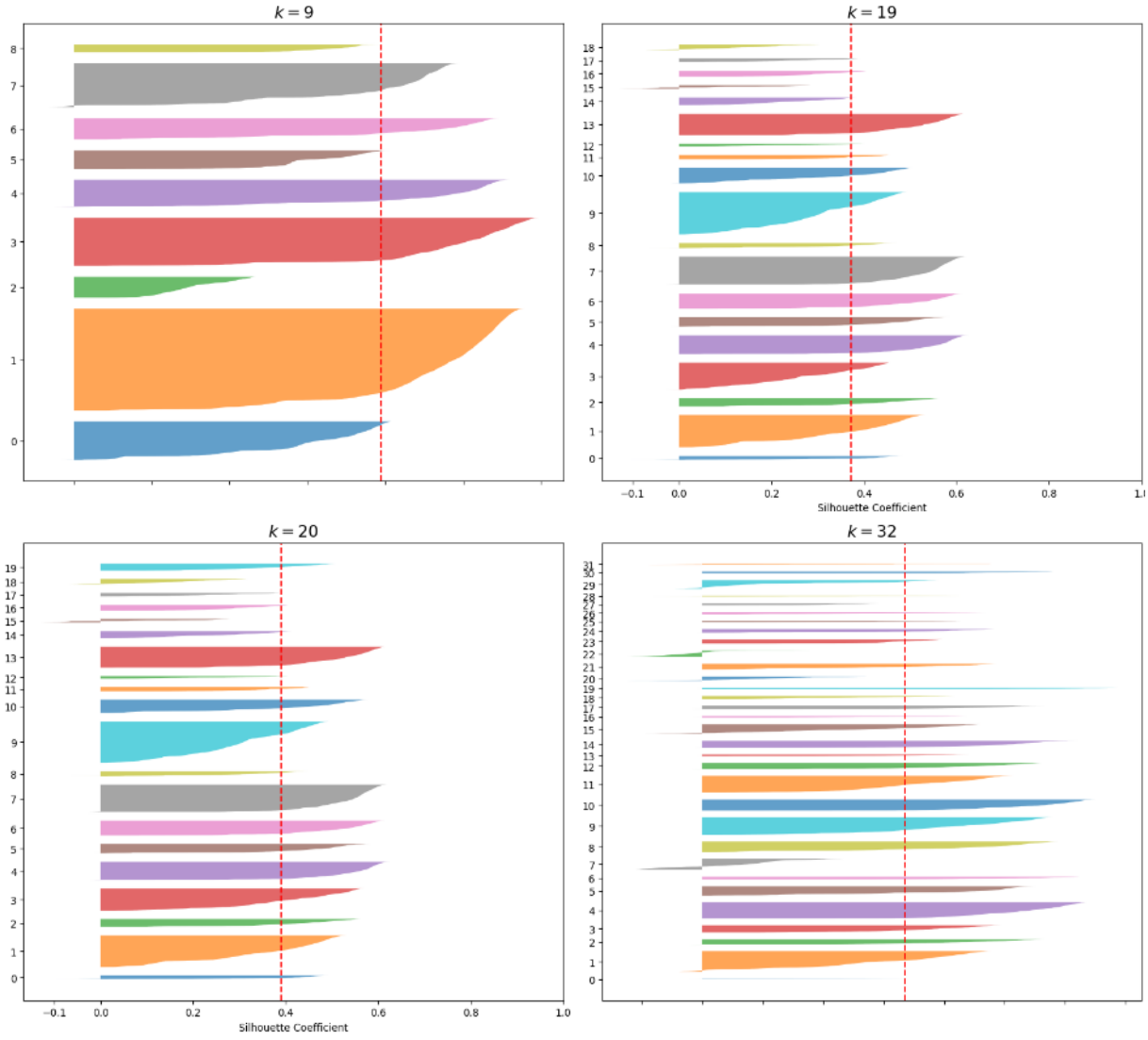
Gráfico del coeficiente de silueta para K-Means con PCA = 15



- **Paso 3:** Diagrama de silueta. Observando los diagramas de silueta para los diferentes valores de k seleccionados (figura 22), se observa que hay muestras mal clasificadas, especialmente en k=32. Además, se nota que para cualquier coeficiente de silueta promedio, ninguno es mayor a 0.4, lo cual está más cercano a 0 que a 1. Por lo tanto, se seleccionó k=20, que es el modelo con menor cantidad de muestras mal clasificadas y el que presenta un mayor coeficiente de silueta promedio.

Figura 22

Diagrama de silueta para para K-Means con PCA = 15




```
# Entrenar el modelo con mejores resultados en este caso kmeans
kmeans4 = KMeans(n_clusters=20)
kmeans4.fit(principal_components15)

# Predicciones de los clusters
y_kmeans4 = kmeans4.predict(principal_components15)
```

las métricas resultantes son:

Silhouette Score: 0.3638793182951564
Calinski Harabasz Score: 2687.272113869127
Davies Bouldin Score: 1.215476643518954
Inertia: 3022.765085801634

Después de entrenar los 4 modelos de k-means con diferentes conjuntos de datos, se compararon las diversas métricas resultantes las cuales se pueden ver en la tabla 9, de esta manera se seleccionó el mejor modelo:

Tabla 9

Tabla de las métricas de calidad de los modelos K-Means

	KMEANS1	KMEANS2	KMEANS3	KMEANS4
Silhouette Score	0.42	0.29	0.53	0.36
Calinski Harabasz Score	3208.47	2933.1	11064.76	2687.27
Davies Bouldin Score	1.08	1.25	0.68	1.22
Inercia	2071.49	5006.55	356.98	3022.77

Silhouette Score: Como se explicó anteriormente, esta métrica cuantifica cuán similar es un punto a su propio clúster en comparación con otros clústeres, tomando valores en el rango [-1, 1].

Calinski Harabasz Score: Esta métrica mide la relación entre la dispersión dentro de los clústeres y la dispersión entre los clústeres. Un valor más alto indica clústeres más densos y bien separados.

Davies Bouldin Score: Esta métrica cuantifica la "compacidad" y la "separación" de los clústeres. Valores más bajos indican clústeres mejor definidos.

Inertia: Como se explicó anteriormente, la inercia es una medida de la coherencia de los clústeres, donde valores más bajos indican clústeres más densos y bien definidos.

Conclusión: Se obtuvieron mejores métricas con el modelo KMeans con 40 clusters, el cual se implementó con un conjunto de datos de 5 componentes (PCA = 5). Esto era lo esperado, ya que en espacios de muy altas dimensiones, las distancias euclidianas tienden a inflarse, fenómeno conocido como la maldición de la dimensionalidad. La ejecución de un algoritmo de reducción de dimensionalidad, como el Análisis de Componentes Principales (PCA), antes de aplicar KMeans puede mitigar este problema y mejorar la calidad de la agrupación.

También se tomó en cuenta el hiperparámetro init='k-means++', el cual inicializa los centroides para que estén (generalmente) distantes entre sí. Sin embargo, este ajuste no mejoró los resultados obtenidos, por lo que se decidió no considerarlo.

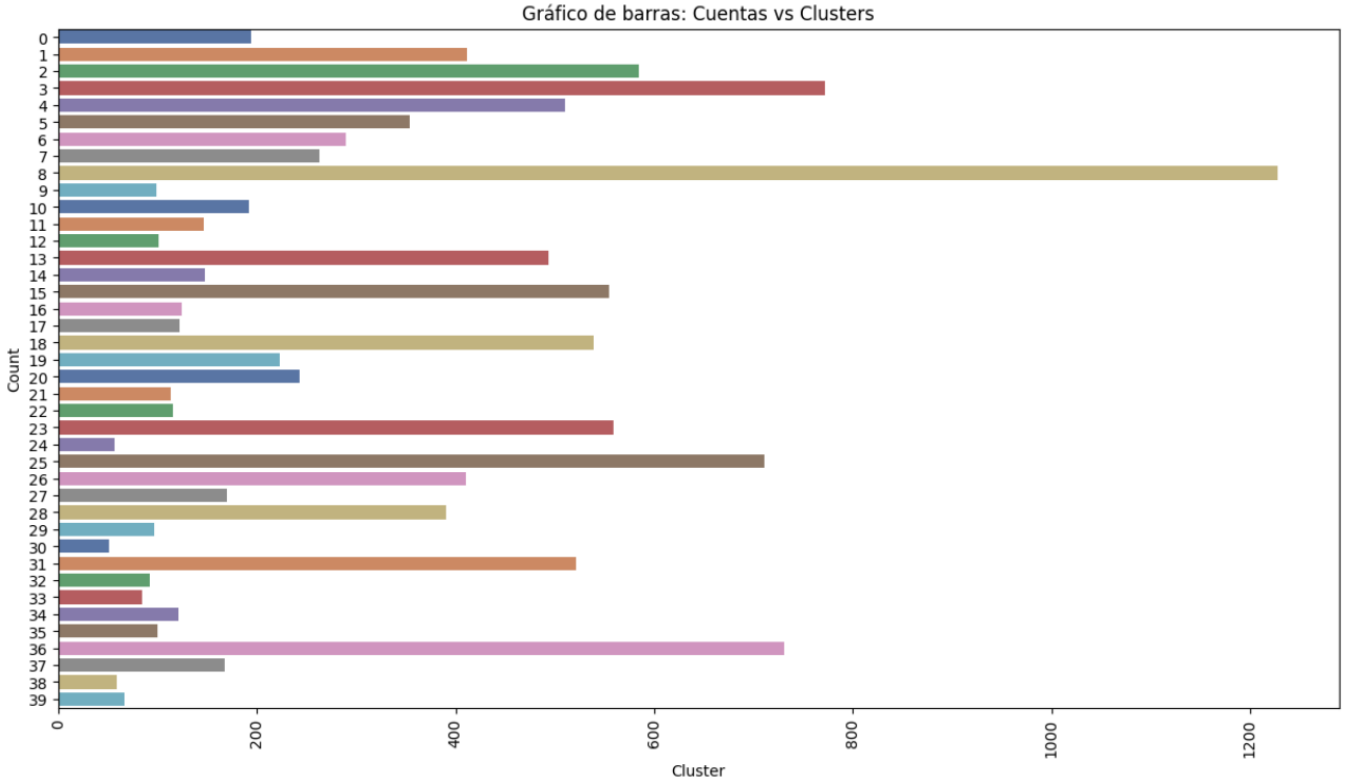
Modelo seleccionado:

```
kmeans3 = KMeans(n_clusters=40)
kmeans3.fit(principal_components5)
```

Riesgo: solo se explica el 81% de la variabilidad de los datos.

Figura 23

Gráfico de barras de la cantidad de muestras en cada cluster



(En el grafico anterior observados la cantidad de muestras que hay en cada clúster)

5.2. Aplicación de DBSCAN

La segunda técnica de clustering implementada fue DBSCAN, la cual realiza agrupaciones basadas en la densidad, identificando regiones de alta densidad separadas por regiones de baja densidad. Este enfoque es completamente diferente al de la técnica anteriormente implementada. Se tuvieron en cuenta las siguientes premisas:

- La densidad de cualquier punto depende del radio especificado, por lo tanto, la definición de este parámetro (eps) es crucial.
- Para cada muestra, el algoritmo cuenta cuántas muestras están dentro de una distancia ε (parámetro eps). Esta región se conoce como el ε-neighborhood.
- Un punto se considera una "instancia central" ("core instance") si tiene al menos un número mínimo de vecinos dentro de una distancia ε, definida por los parámetros epsilon y min_samples. Estas "instancias centrales" se encuentran en regiones densas de datos.
- Todas las muestras dentro de la vecindad de una "instancia central" pertenecen al mismo grupo.
- Las muestras que no son "instancias centrales" y no tienen ninguna "instancia central" en su vecindad se consideran anomalías (estas muestras tienen un índice

de clúster igual a -1 según el algoritmo).

- Los grupos identificados por DBSCAN pueden tener cualquier forma.
- DBSCAN puede enfrentar dificultades cuando los grupos tienen densidades muy variables, así como en conjuntos de datos de alta dimensión, donde la densidad es más difícil de definir.
- DBSCAN puede ser computacionalmente costoso cuando se necesitan calcular todas las proximidades por pares, como es común en datos de alta dimensión.
- El parámetro min_samples controla principalmente la tolerancia del algoritmo hacia el ruido. En conjuntos de datos grandes y ruidosos, como en nuestro caso, puede ser recomendable aumentar este parámetro.
- Se recomienda trabajar con datos estandarizados y normalizados para este modelo.
- La estandarización ayuda a manejar variables en diferentes escalas y preserva la distribución original de los datos, lo cual es crucial para interpretar correctamente los resultados.
- Se sugiere elegir el parámetro min_samples de acuerdo con el número de atributos P de la base de datos, utilizando la fórmula: min_samples = 2 * P.

Se realizó un procedimiento similar al análisis de variabilidad de los datos aplicando una transformación y reducción mediante componentes principales, pero esta vez se aplicó a los datos estandarizados.

Se procedió a estandarizar los datos:

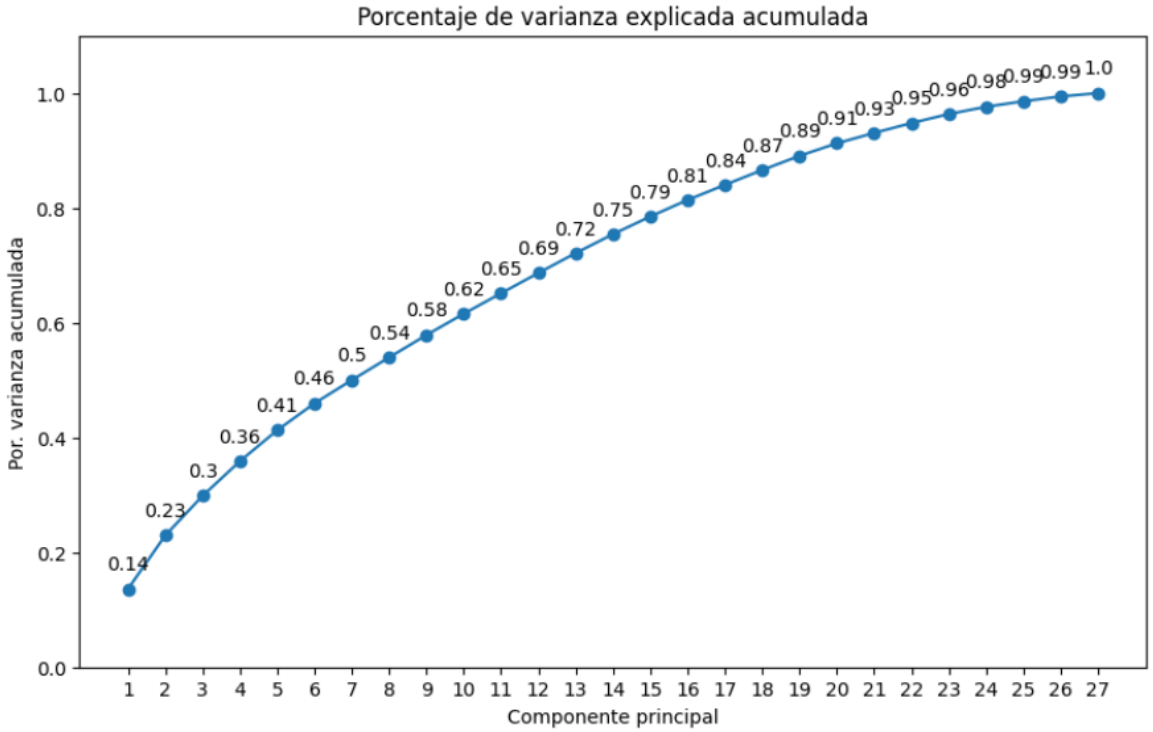
```
scaler = StandardScaler()
scaler.fit(X)
x_scaled = scaler.transform(X)
```

se realizó el análisis de variabilidad:

```
x1 = x_scaled.copy()
pca =  PCA(n_components = 27)
pca.fit(x1)
```

Figura 24

Varianza explicada acumulada del PCA con los datos originales estandarizados



Al observar el gráfico del porcentaje de varianza explicada por las nuevas componentes al aplicar PCA a los datos estandarizados, se notó que se requerían más componentes para explicar la varianza de los datos en comparación con la normalización. Por lo tanto, se decidió aplicar ambos modelos con las respectivas transformaciones para realizar comparaciones basadas en métricas y tomar una decisión sobre cuál transformación utilizar.

Inicialmente, se entrenaron dos modelos: uno con los datos estandarizados y otro con los datos normalizados. Luego, se aplicó PCA con 20 componentes a los datos estandarizados (explicando el 90% de la varianza, como se muestra en la figura 24), y PCA con 8 componentes a los datos normalizados (explicando el 91% de la varianza, figura 10).

Se llevaron a cabo pasos adicionales para determinar los valores óptimos de min_samples y epsilon, ya que son hiperparámetros clave para la generación de los clusters.

5.2.1. Modelo 1: DBSCAN con datos estandarizados

- **Paso 1:** Se calculó el min_sample=2*27=54

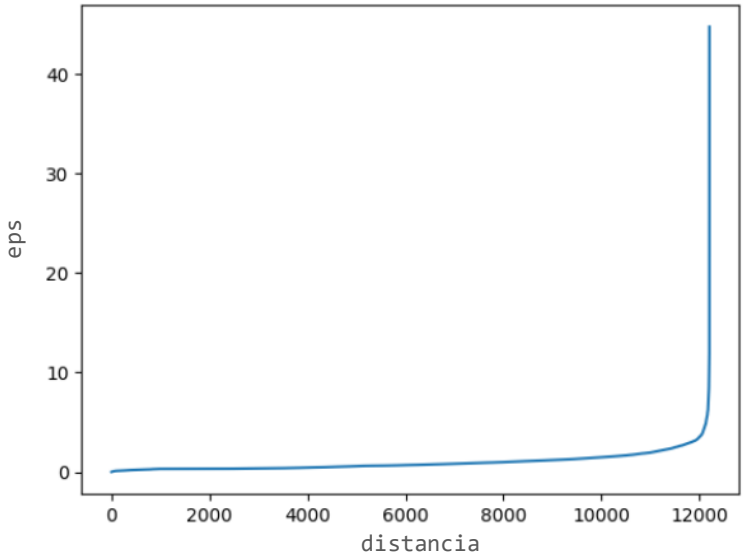
- **Paso 2:** Se entrenó un algoritmo de los vecinos más cercanos. Solo se calculan las distancias entre las muestras y se organizan en orden ascendente para generar una gráfica que ayude a determinar el valor de épsilon (figura 25 y 26), buscando el codo en la curva. Este punto representa donde la distancia entre los puntos vecinos comienza a aumentar significativamente.

```
neighbors = NearestNeighbors(n_neighbors=min_samples)
```

```
neighbors_fit = neighbors.fit(x_scaled)
distances, indices = neighbors_fit.kneighbors(x_scaled)
```

Figura 25

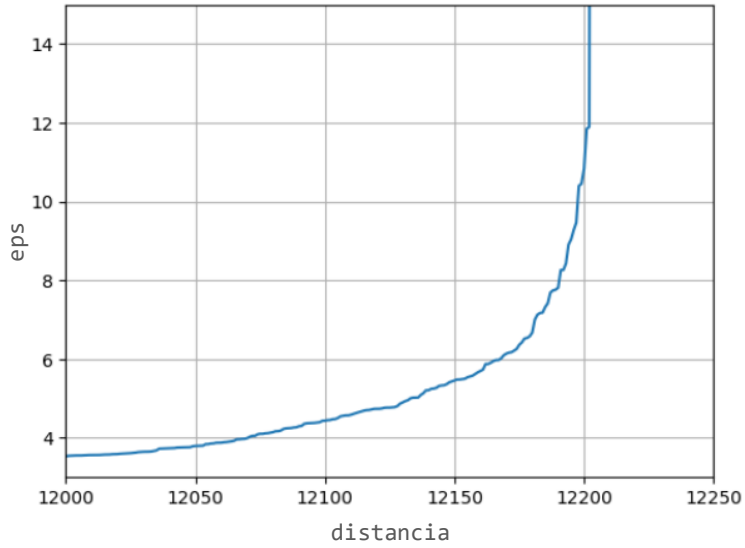
Gráfico 1 del codo para DBSCAN con los datos estandarizados (distancia vs épsilon)



Este es un primer acercamiento para determinar un valor óptimo de epsilon. Por lo tanto, se definió un intervalo desde 5 hasta 14 para observar el comportamiento en los siguientes pasos.

Figura 26

Gráfico 2 del codo para DBSCAN con los datos estandarizados (distancia vs épsilon)

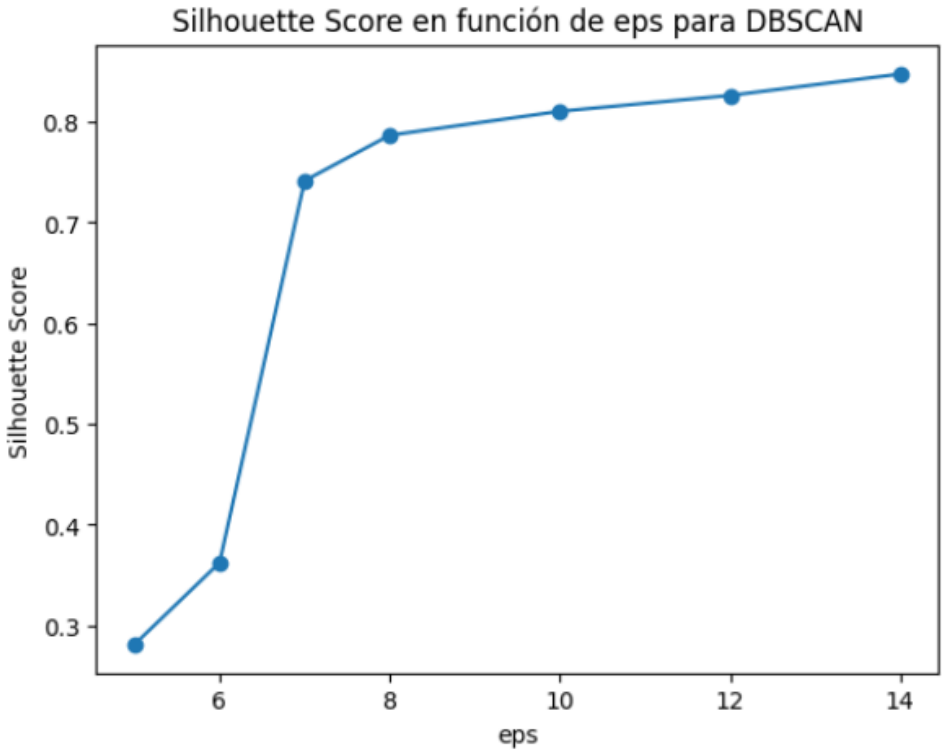


- **Paso 3:** Se calculó el score de silueta en función de épsilon (figura 27). El

parámetro épsilon (ε) define la distancia máxima entre dos puntos para que se consideren vecinos directos en un grupo. Es un factor crítico que determina la forma en que se agrupan los datos y puede tener un impacto significativo en los resultados del algoritmo DBSCAN. Por otro lado, el coeficiente de silueta es una medida de la cohesión y separación de los grupos obtenidos mediante un algoritmo de agrupamiento. Este coeficiente evalúa qué tan similar es un punto a su propio grupo en comparación con otros grupos vecinos. Aunque el coeficiente de silueta no se utiliza directamente para ajustar el valor de épsilon en DBSCAN, puede ser útil para evaluar la calidad general de los grupos generados por diferentes valores de épsilon. Si bien el coeficiente de silueta no siempre es aplicable en el contexto de DBSCAN debido a su naturaleza basada en densidad, aún puede proporcionar una indicación general de la cohesión de los grupos.

Figura 27

Gráfico del coeficiente de silueta para DBSCAN con los datos estandarizados



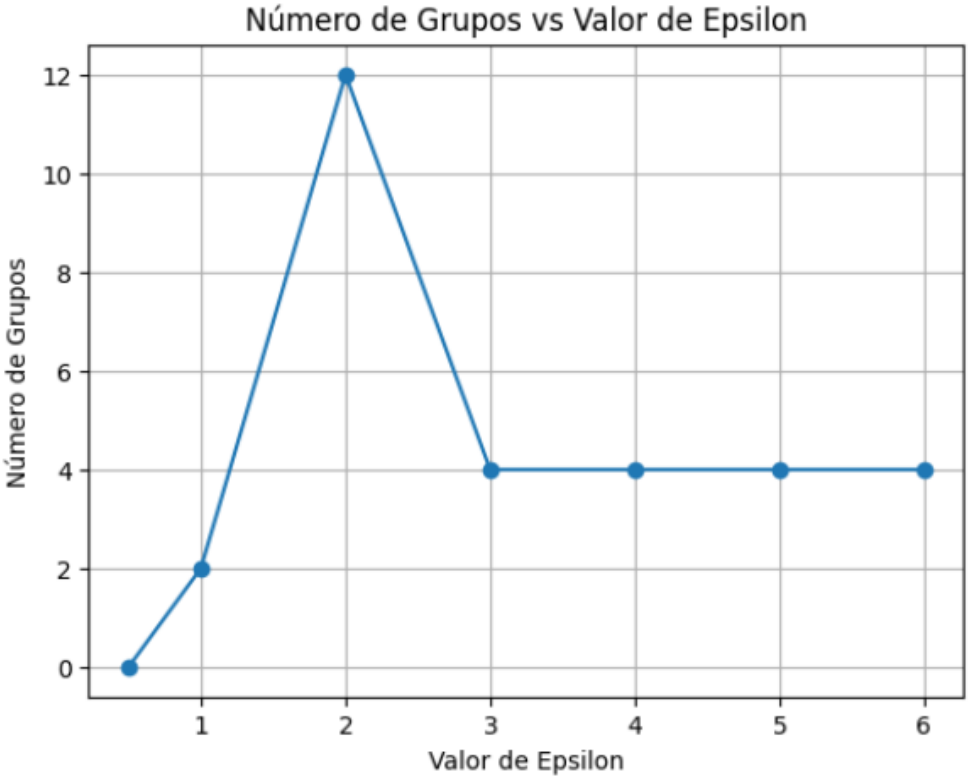
Para los valores seleccionados de epsilon (ε), se observa que el score de silueta presenta un comportamiento creciente a medida que se aumenta el valor de este parámetro.

Paso 4: se determinó el número de grupos a partir de épsilon. Se calculó el número de grupos óptimos por diferentes valores de épsilon, ayudando a identificar un valor óptimo de épsilon de acuerdo con la cantidad de grupos (ver figura 28), en este caso vemos con un valor de eps=2 podemos obtener 12 grupos,

lo cual sigue siendo una cantidad muy pequeña de clúster de acuerdo con la solución esperada.

Figura 28

Gráfico del número de grupos vs. el valor de Épsilon



- **Paso 5:** Calculo de la densidad media. La densidad de un grupo se refiere a cuántos puntos están cercanos entre sí dentro de ese grupo. La densidad media de los grupos es una medida que evalúa qué tan densos son los puntos dentro de los grupos identificados por el algoritmo de agrupamiento DBSCAN en relación con un valor dado de epsilon. En esencia, busca cuantificar qué tan "compactos" o "juntos" están los puntos dentro de cada grupo. Para comprender mejor qué mide y a qué presta atención la densidad media de los grupos, es útil tener en cuenta algunos conceptos clave:

    - DBSCAN y el parámetro épsilon (ε): DBSCAN es un algoritmo de agrupamiento basado en la densidad que clasifica los puntos en tres categorías: puntos centrales, puntos fronterizos y ruido. El parámetro épsilon (ε) en DBSCAN define la distancia máxima entre dos puntos para que se consideren vecinos directos en un grupo. Esto significa que todos los puntos dentro de una distancia ε del punto central se consideran parte del mismo grupo.

    - Densidad de un grupo: En DBSCAN, la densidad de un grupo se refiere a cuántos puntos se encuentran dentro de una distancia ε de cualquier punto

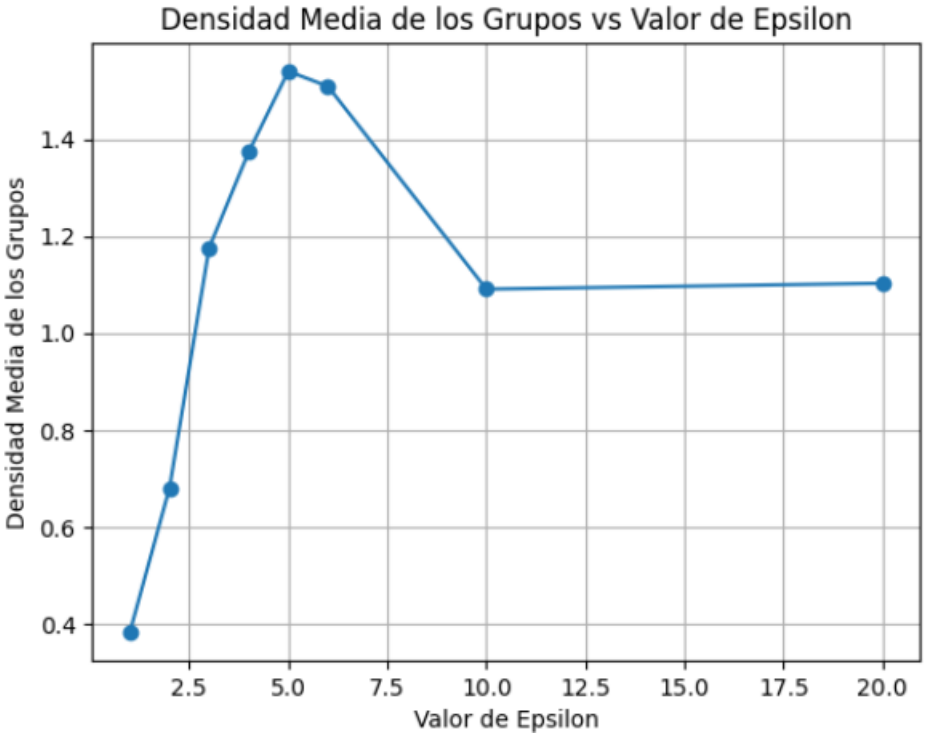
central dentro del grupo. Cuantos más puntos haya dentro de esta distancia, mayor será la densidad del grupo.

- Densidad media de grupos: La densidad media de grupos es simplemente el promedio de las densidades de todos los grupos identificados por DBSCAN para un valor específico de épsilon. Es una medida que proporciona una indicación general de cuán densos son los grupos en relación con el valor de épsilon dado.

Entonces, la densidad media de grupos analiza cómo varía la densidad de los grupos identificados por DBSCAN a medida que se ajusta el valor de épsilon. Si la densidad media de grupos es alta, significa que los puntos dentro de los grupos están relativamente cerca unos de otros, lo que sugiere grupos compactos y bien definidos. Por otro lado, si la densidad media de grupos es baja, puede indicar que los grupos son dispersos o que los puntos dentro de ellos están más separados, lo cual podría afectar la interpretación de los resultados del agrupamiento. En resumen, la densidad media de grupos proporciona información útil para evaluar la cohesión de los grupos identificados por DBSCAN en función de diferentes valores de épsilon.

Figura 29

Gráfico del coeficiente de densidad media de los grupos vs. Valor de Épsilon



Observando la gráfica de la densidad media (figura 29), vimos que tiende a crecer a medida que aumentamos el valor de epsilon hasta alcanzar un máximo en 5, y luego decrece hasta aproximadamente 10, manteniéndose luego

constante. Por lo tanto, el valor de epsilon que maximiza la densidad media es epsilon = 5. En consecuencia, se entrenó el modelo utilizando este valor.

```
dbscan1 = DBSCAN(eps=5,min_samples=54)

# Aplicar DBSCAN a los datos
clusters = dbscan1.fit_predict(x_scaled)
```

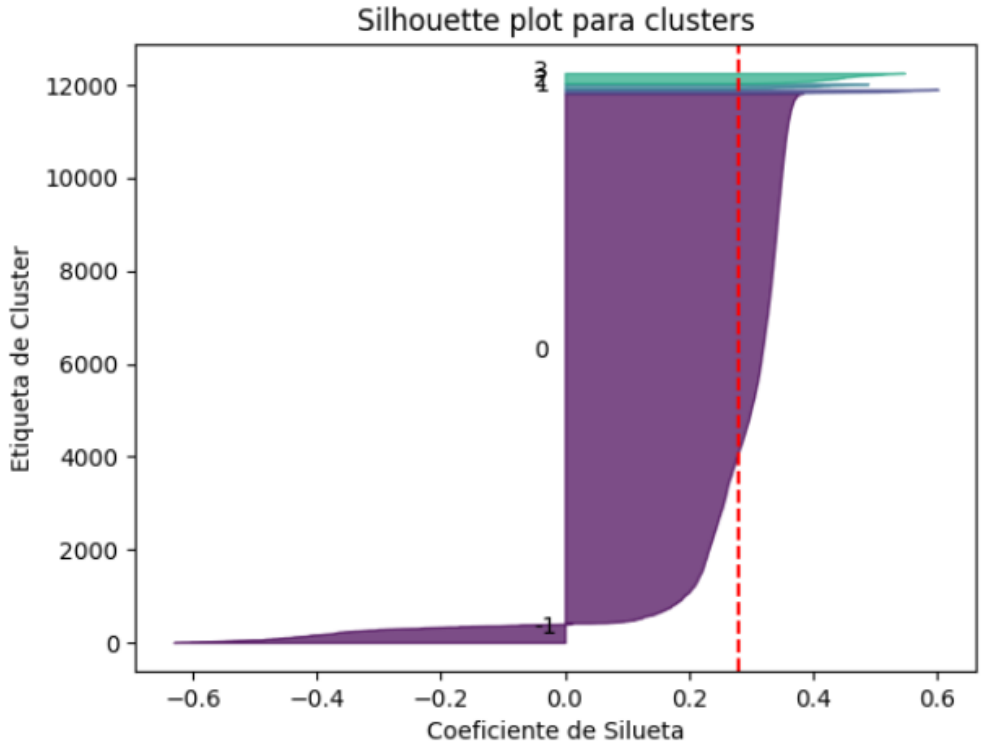
Las métricas resultantes son:

```
Silhouette Score: 0.28132215877572203
Davies-Bouldin Score: 2.437855415943665
Calinski Harabasz Score: 373.54257690178554
```

Después de observar las métricas resultantes del modelo implementado con los hiperparámetros óptimos, encontramos que los resultados no son satisfactorios. Al analizar el diagrama de silueta de la figura 30, observamos que la mayoría de los puntos fueron agrupados dentro del clúster 0. Además, solo se obtuvieron 5 clústeres, lo cual no cumple con el objetivo del agrupamiento.

Figura 30

Diagrama de silueta para para DBSCAN con los datos estandarizados



En este caso, el valor de épsilon recomendado según la densidad media no fue el adecuado. Este resultado podría deberse a que muchas muestras están agrupadas en el clúster 0, lo que aumenta la densidad media al promediar con otras densidades, pero resulta en una distribución muy pobre entre los grupos (grupos no homogéneos).

Cuando el agrupamiento genera más grupos, estos están mejor distribuidos como se puede ver en la figura 31, lo cual es coherente con la solución del problema. Esto ocurre cuando se reduce el valor de épsilon (ε = 2), pero muchas muestras quedan en el clúster -1 (no clasificado), lo cual afecta negativamente la densidad media y se refleja en métricas pobres en cuanto a la calidad de los clústeres, como se observa a continuación:

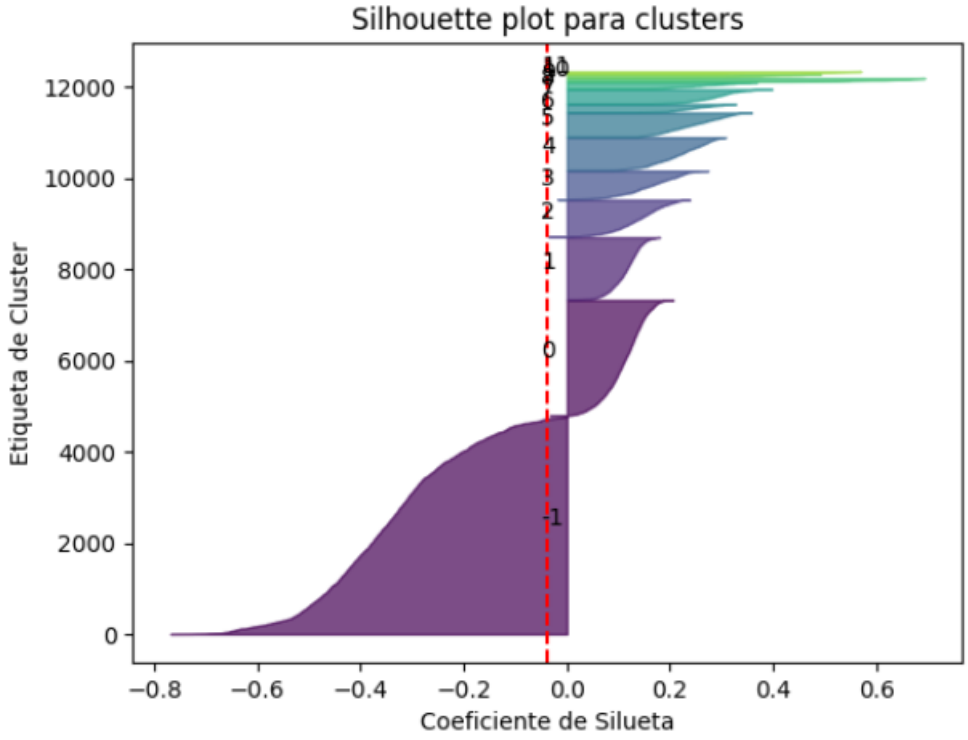
```
dbscan1_2 = DBSCAN(eps=2,min_samples=54)

# Aplicar DBSCAN a los datos
clusters = dbscan1_2.fit_predict(x_scaled)
```

Las métricas resultantes:

```
Silhouette Score: -0.03750315147532369
Davies-Bouldin Score: 2.467169857829629
Calinski Harabasz Score: 168.65624705127848
```

Figura 31

Diagrama de silueta para DBSCAN con los datos estandarizados



De acuerdo con lo anterior, se concluye que los modelos entrenados con los datos estandarizados (sin reducir la dimensionalidad) no proporcionan resultados adecuados para la solución del problema. Por lo tanto, se procede a realizar el entrenamiento con los datos normalizados.

5.2.2. Modelo 2: DBSCAN con datos normalizados

- **Paso 1:** Se calculó el min_sample=2*27=54
- **Paso 2:** Se entrenó un algoritmo de los vecinos más cercanos (figura 32 y 33).

Figura 32

Gráfico 1 del codo para DBSCAN con los datos normalizados (distancia vs épsilon)

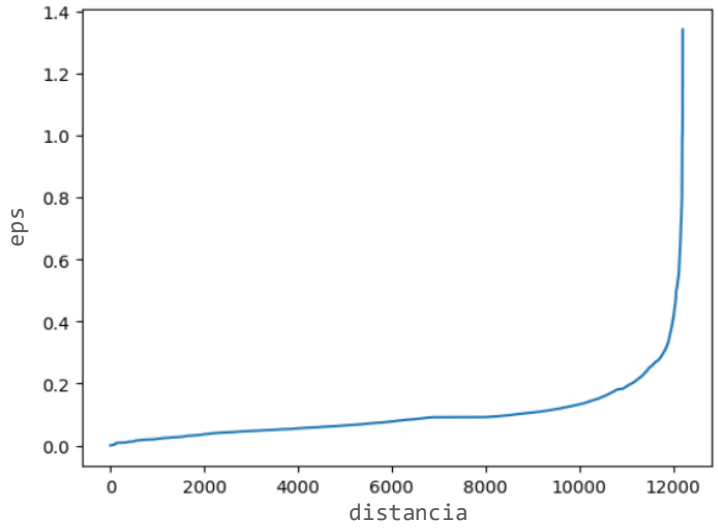
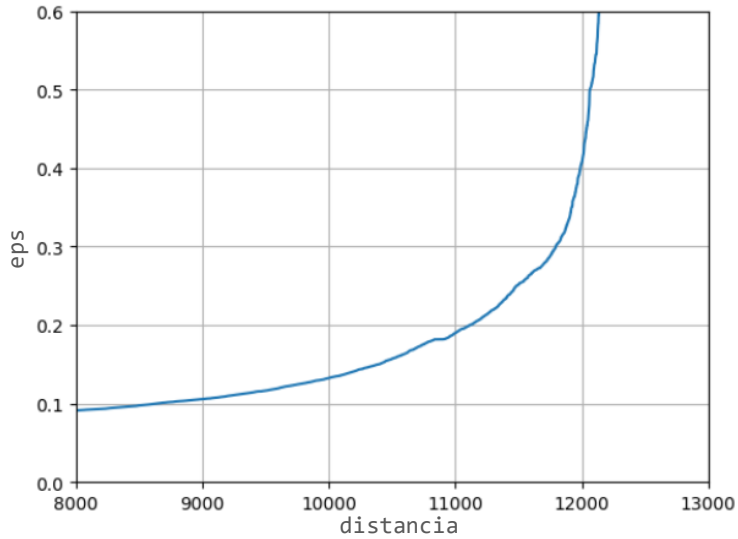


Figura 33

Gráfico 1 del codo para DBSCAN con los datos normalizados (distancia vs épsilon)

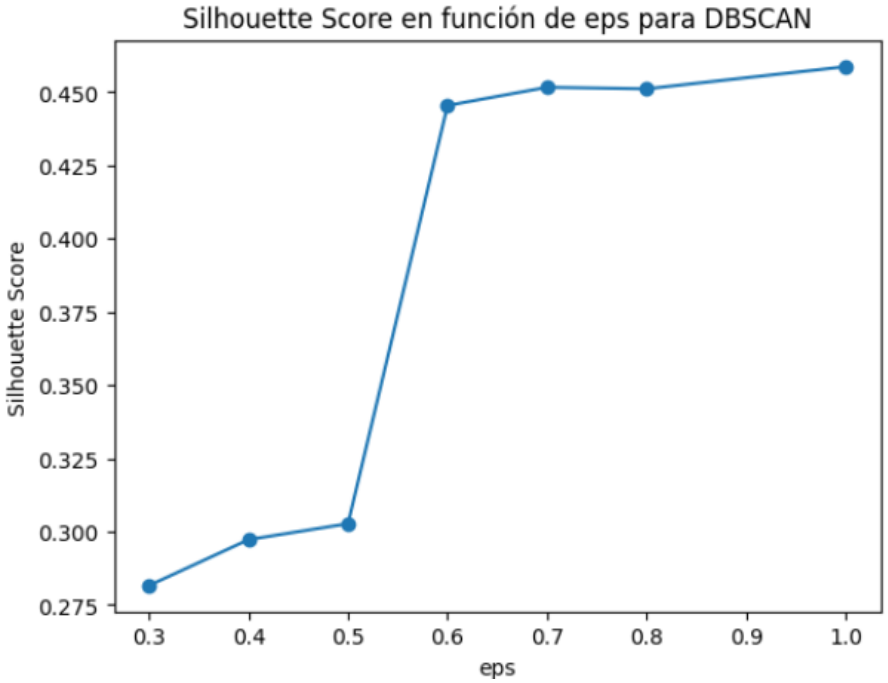


Según la gráfica, el punto donde la distancia entre los puntos vecinos comienza a aumentar significativamente es aproximadamente en epsilon (ε) = 0.3. Para los siguientes pasos, se tomará un rango desde 0.3 hasta 1.

- **Paso 3:** Se calculó el score de silueta en función de épsilon.

Figura 34

Gráfico del coeficiente de silueta para DBSCAN con los datos normalizados

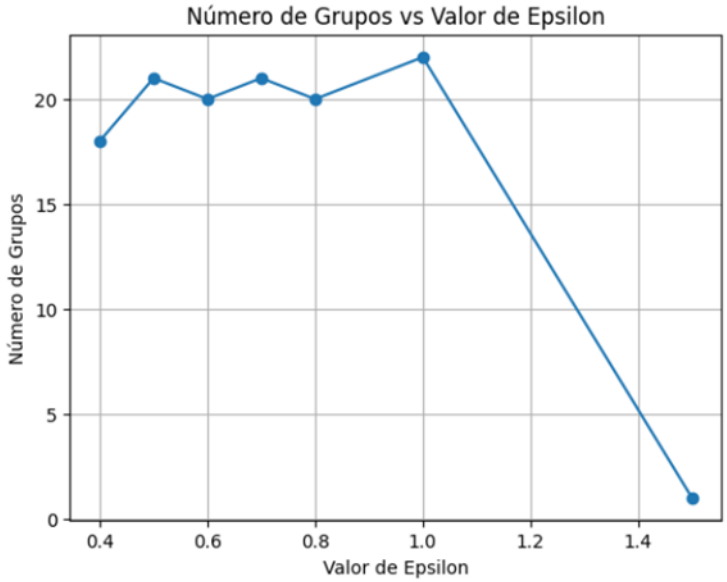


En la figura 34 pudimos observar que el valor del score de silueta crece significativamente cuando epsilon (ε) toma un valor de 0.6, y luego de manera más gradual.

- **Paso 4:** Se determinó el número de grupos a partir de épsilon

Figura 35

Gráfico del número de grupos vs. el valor de Épsilon

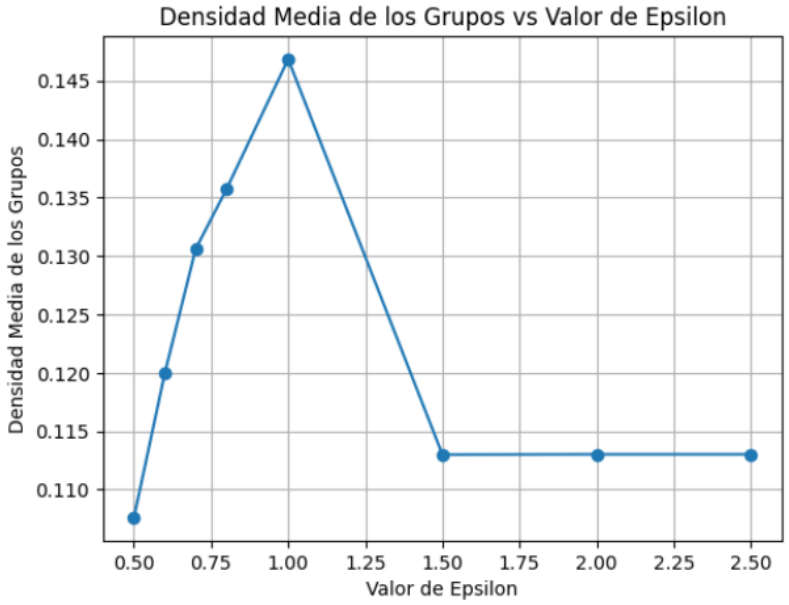


En la figura 35 se observa que con valores de épsilon entre 0.5 y 1 se forman entre 20 y 22 clústeres, pero cuando el valor de épsilon es mayor a 1, el número de clústeres disminuye rápidamente.

- **Paso 5:** Calculo de la densidad media.

Figura 36

Gráfico del coeficiente de densidad media de los grupos vs. Valor de Épsilon



Observando la gráfica (figura 36), se puede ver claramente que cuando el radio de búsqueda toma un valor de 1, presenta una mayor densidad media. También para este valor de épsilon se forman 22 grupos, sin contar el clúster -1, lo cual está alineado con la solución del problema. Además, entre los valores analizados, presenta el mayor score de silueta. Por lo tanto, este valor se selecciona para realizar el entrenamiento del modelo.

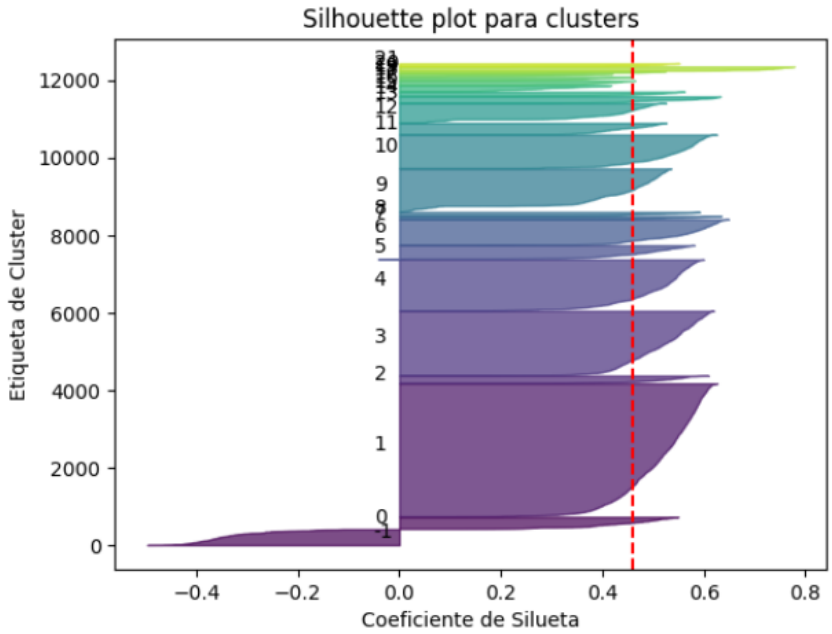
```
dbscan2 = DBSCAN (eps=1, min_samples=54)

 # Aplicar DBSCAN a los datos
clusters = dbscan2.fit_predict(scaled_data)
```

Las métricas resultantes:

```
Silhouette Score: 0.45862381905770205
Davies-Bouldin Score: 1.5056014943717597
Calinski Harabasz Score: 1956.4282753690404
```

Figura 37

Diagrama de silueta para DBSCAN con los datos normalizados



Se evidenció que los clústeres no son muy homogéneos (ver figura 37), pero están mejor distribuidos que en los modelos anteriores. También se observa la presencia de varios datos considerados como anomalías. De acuerdo con esto, se aplicará la reducción de dimensionalidad con PCA para ambas transformaciones (estandarización y normalización), con la expectativa de obtener mejores resultados con una dimensionalidad reducida. Se anticipa que las componentes derivadas del conjunto de datos normalizados serán superiores a las obtenidas con los datos estandarizados.

5.2.3. Modelo 3: DBSCAN con PCA=20 - datos estandarizados

Se aplicó PCA para los datos estandarizados con 20 componentes.

```
# Aplicar PCA con 20 componentes datos estandarizados
pca =  PCA(n_components = 20)
principal_components20 = pca.fit_transform(x_scaled)
```

- **Paso 1**: Se calculó el min_sample=2*20=40
- **Paso 2**: Se entrenó un algoritmo de los vecinos más cercanos.

Figura 38

Gráfico 1 del codo para DBSCAN con PCA=20 datos estandarizados (distancia vs épsilon)

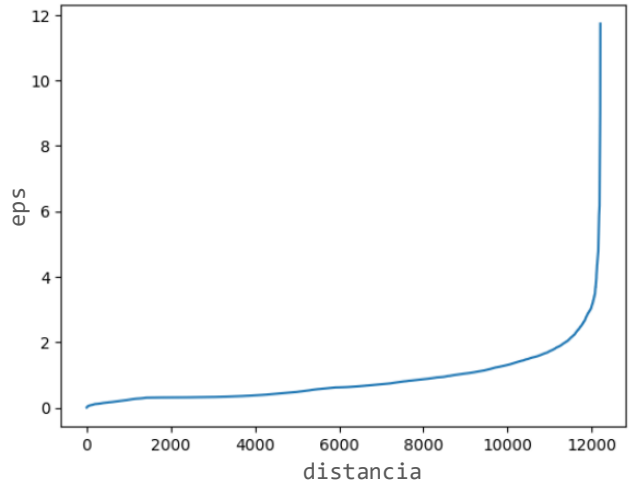
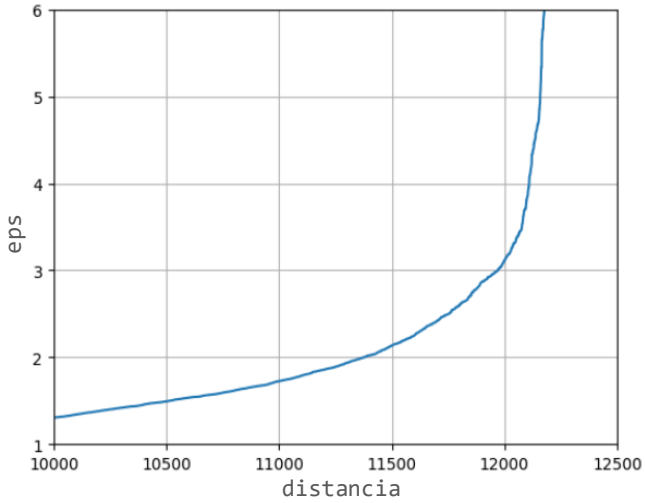


Figura 39

Gráfico 2 del codo para DBSCAN con PCA=20 datos estandarizados (distancia vs épsilon)

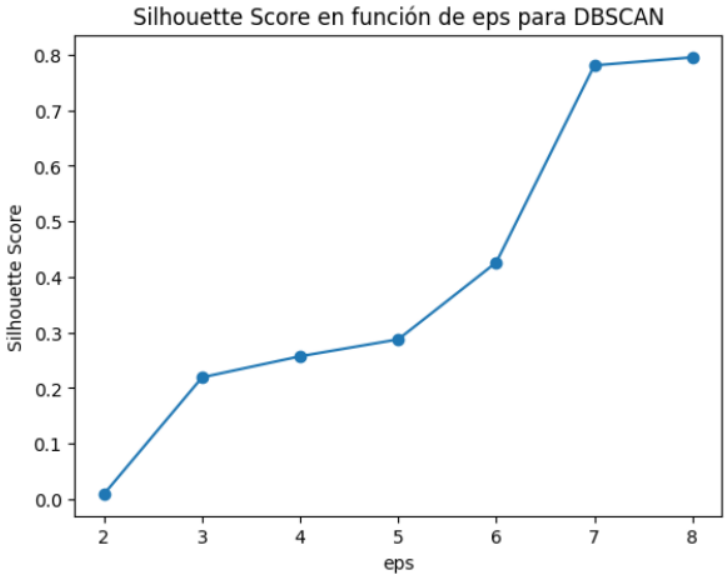


Según la gráfica de las figuras 38 y 39, el punto donde la distancia entre los puntos vecinos comienza a aumentar significativamente es aproximadamente en (ε) = 3. Para los siguientes pasos, se considera un rango de 2 hasta 8.

- **Paso 3:** Se calculó el score de silueta en función de épsilon.

Figura 40

Gráfico del coeficiente de silueta para DBSCAN con PCA=20 datos estandarizados

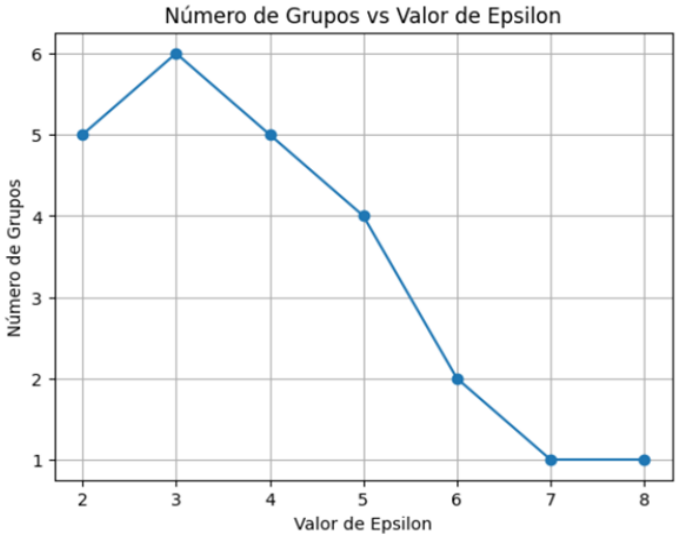


Se observa en la figura 40 que el score de silueta crece a medida que el radio de búsqueda aumenta; específicamente, cuando épsilon toma valores mayores a 6, el score de silueta aumenta más rápidamente.

- **Paso 4:** Se determinó el número de grupos a partir de épsilon

Figura 41

Gráfico del número de grupos vs. el valor de Épsilon

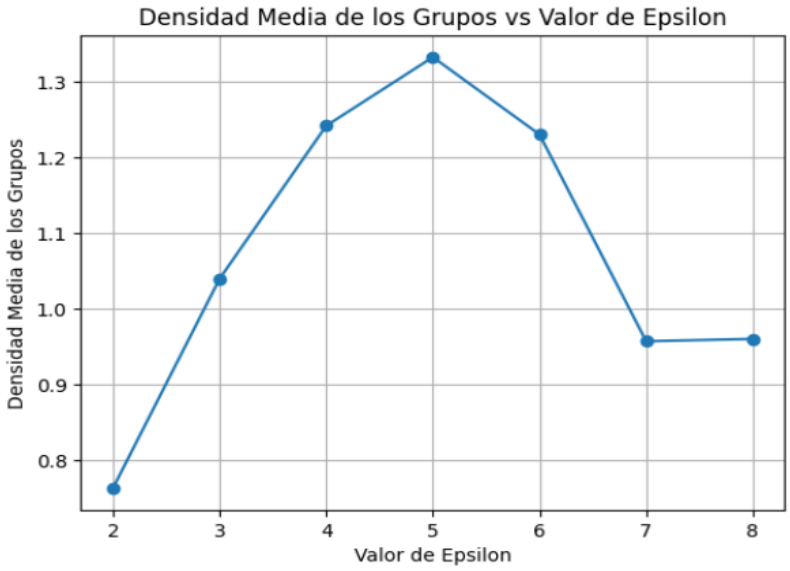


Se observa que para cualquiera de los valores de épsilon analizados se forman pocos grupos (ver figura 41).

- **Paso 5:** Calculo de la densidad media.

Figura 42

Gráfico del coeficiente de densidad media de los grupos vs. Valor de Épsilon



Pudimos evidenciar que el valor que maximizó la densidad media es cuando épsilon tiene un valor de 5 (figura 42). Sin embargo, este valor solo forma 5 clústeres, incluyendo el grupo -1, y presenta un score de silueta muy bajo, lo que indica que la calidad de los clústeres es baja.

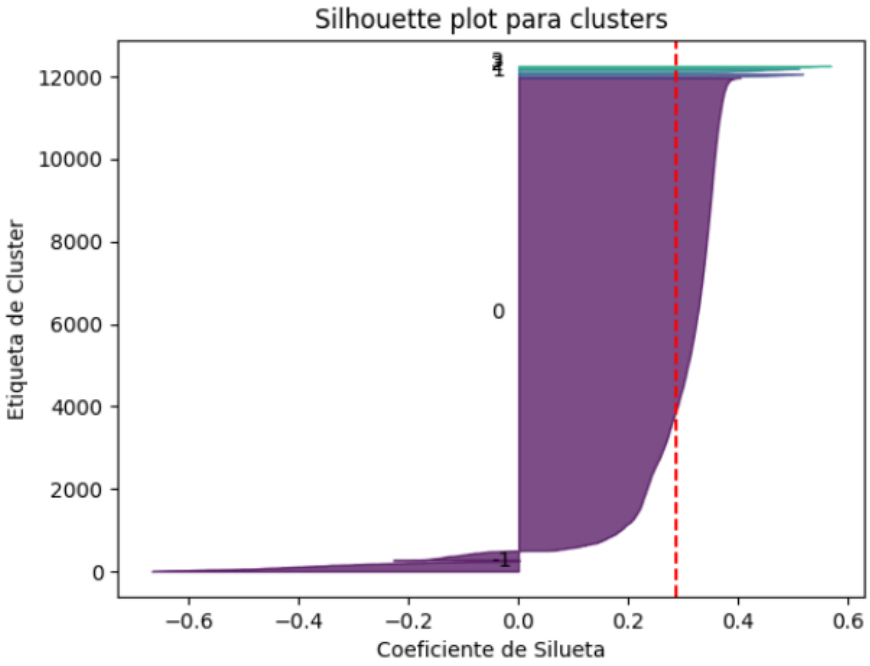
```
dbscan3 = DBSCAN(eps=5 ,min_samples=40)

# Aplicar DBSCAN a los datos
    clusters = dbscan3.fit_predict(principal_components20)
```

Las métricas resultantes:

```
Silhouette Score: 0.28749903384364234
Davies-Bouldin Score: 2.305823986378384
Calinski Harabasz Score: 299.99059890436365
```

Figura 43

Diagrama de silueta para DBSCAN con PCA=20 datos estandarizados



Finalmente, se observó el diagrama de silueta (figura 43) corroborando todo lo mencionado anteriormente. No se obtienen buenos resultados al aplicar DBSCAN con PCA=20 para los datos estandarizados, que explican el 90% de la variabilidad de los datos. El valor de épsilon se seleccionó según la densidad media de los clústeres, pero se evidencia que este resultado está relacionado con la agrupación mayoritaria en el clúster 0, además de que la cantidad de grupos no cumple con el objetivo deseado.

5.2.4. Modelo 4: DBSCAN con PCA=7 - datos normalizados

Se aplicó PCA para los datos estandarizados con 20 componentes.

```
# Aplicar PCA con 7 componentes datos normalizados
pca = PCA(n_components = 7)
principal_components7 = pca.fit_transform(x5)
```

- **Paso 1:** Se calculó el min_sample=2*7=14
- **Paso 2:** Se entrenó un algoritmo de los vecinos más cercanos.

Figura 44

Gráfico 1 del codo para DBSCAN con PCA=7 datos normalizados (distancia vs épsilon)

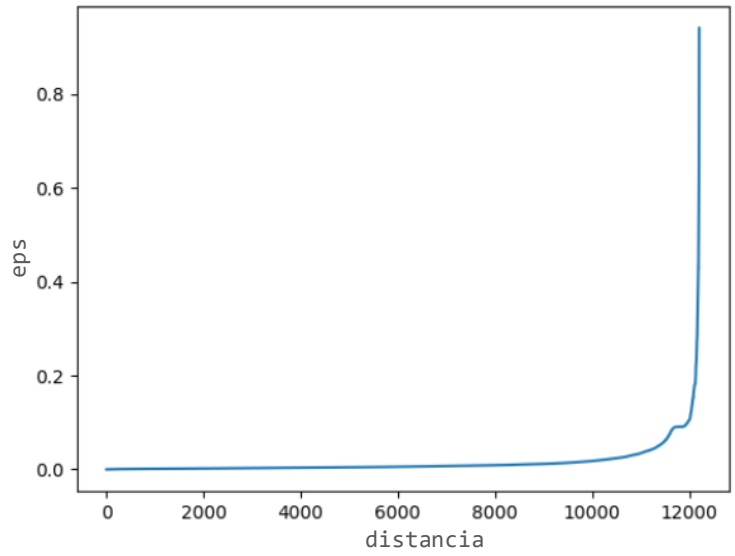
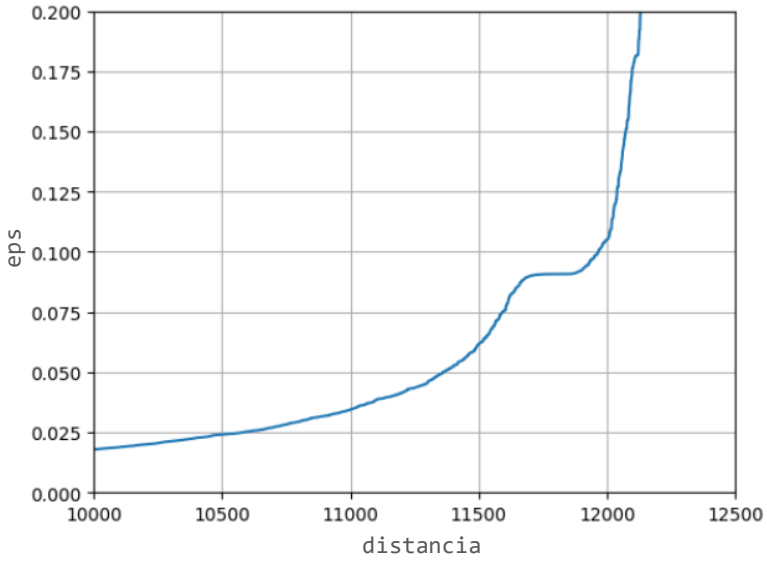


Figura 45

Gráfico 2 del codo para DBSCAN con PCA=7 datos normalizados (distancia vs épsilon)

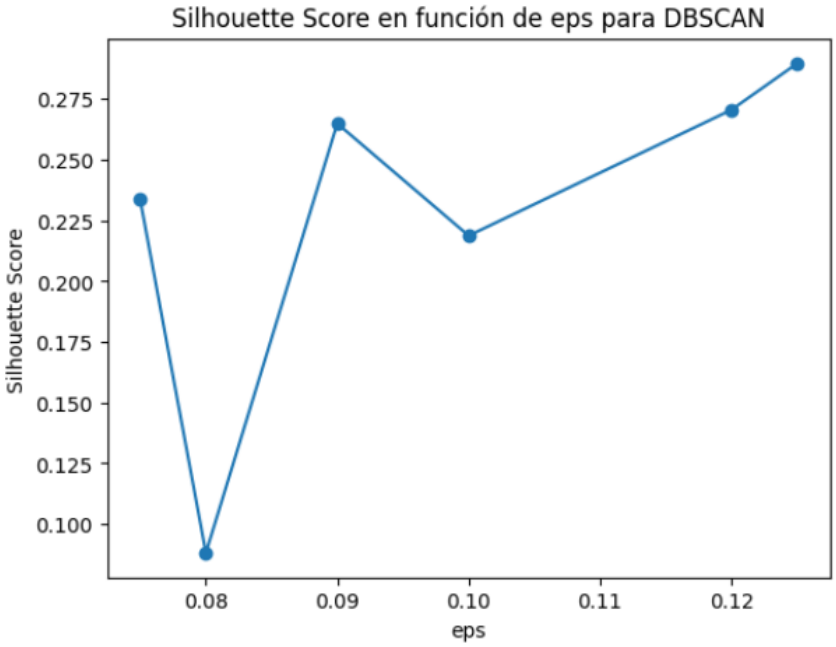


Según las figuras 44 y 45, el punto donde la distancia entre los puntos vecinos comienza a aumentar significativamente es aproximadamente en eps=0.080. Para los demás pasos, se tomará un rango de 0.075 hasta 0.125.

- **Paso 3:** Se calculó el score de silueta en función de épsilon.

Figura 46

Gráfico del coeficiente de silueta para DBSCAN con PCA=7 datos normalizados

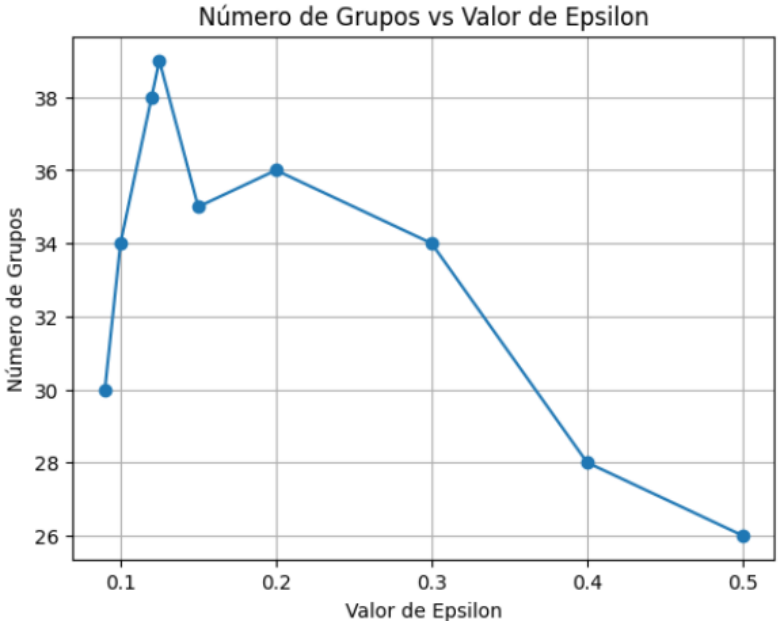


Se evidencia en la figura 46 que para cualquier valor de épsilon analizado se presentan scores de silueta muy bajos (más cercanos a 0 que a 1).

- **Paso 4:** Se Determinó el número de grupos a partir de épsilon
-

Figura 47

Gráfico del coeficiente de silueta para DBSCAN con PCA=7 datos normalizados



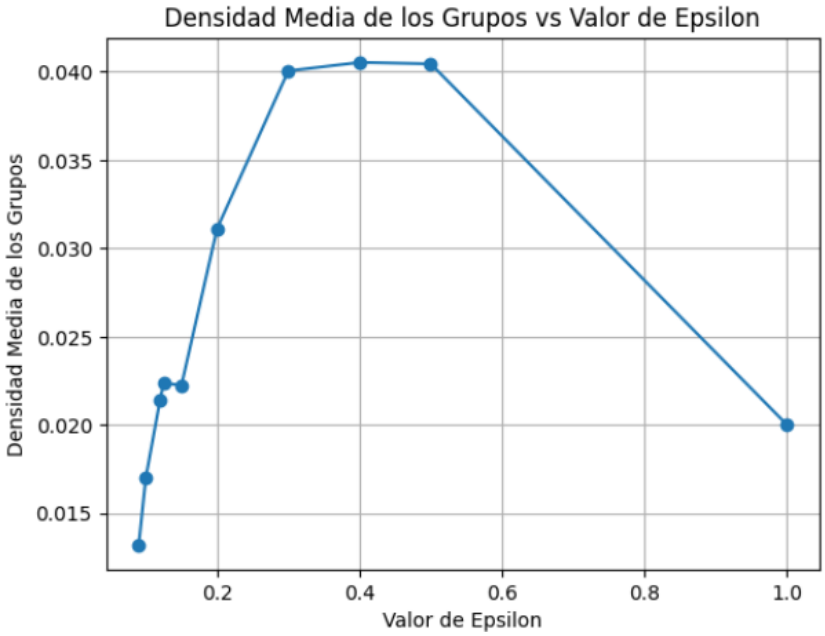
Podemos observar en la figura 47 que para los rangos de búsqueda analizados se forman entre 26 y 39 clústeres, lo cual representa un número adecuado de grupos

para abordar el problema.

- **Paso 5:** Calculo de la densidad media.

Figura 48

Gráfico del coeficiente de densidad media de los grupos vs. Valor de Épsilon



En la figura 48 se observó que el valor que maximiza la densidad media es cuando épsilon tiene un valor de 0.4. Además, con este radio de búsqueda se forman 29 clústeres, por lo tanto, se selecciona este valor para entrenar el modelo con las 7 componentes de los datos normalizados.

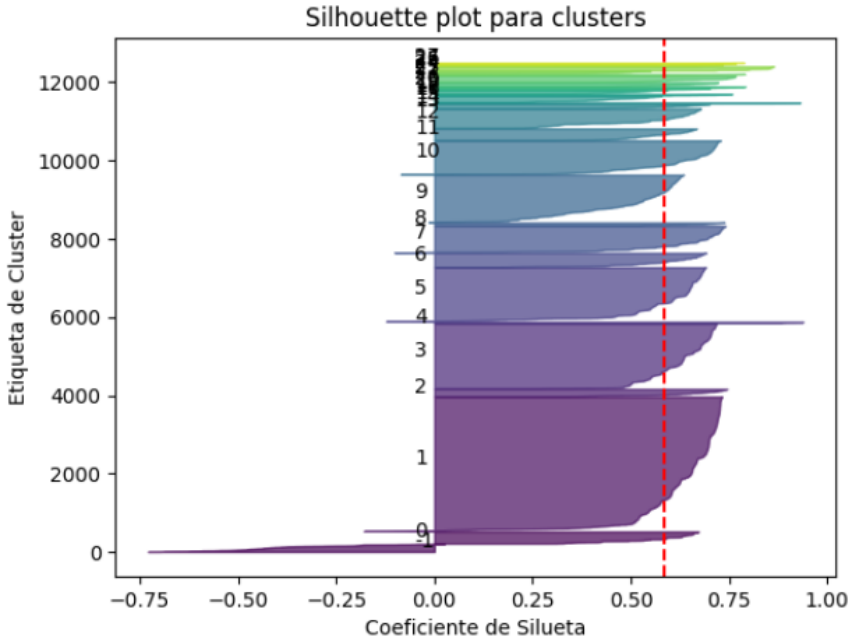
```
dbscan4 = DBSCAN(eps=0.4, min_samples=14)

# Aplicar DBSCAN a los datos
clusters_dbscan4 = dbscan4.fit_predict(principal_components7)
```

Las métricas resultantes:

```
Silhouette Score: 0.5853958405788606
Davies-Bouldin Score: 1.1141313681029525
Calinski Harabasz Score: 2987.1836262680345
```

Figura 49

Diagrama de silueta para DBSCAN con PCA=7 datos estandarizados



Se observó el diagrama de silueta (figura 49), donde se evidenció que los grupos no son homogéneos en cuanto a la cantidad de muestras, con una alta concentración en el clúster 1. Esto podría explicar por qué el valor de la densidad media fue más alto que para los otros valores de épsilon analizados anteriormente. Aun así, este modelo es el que mejor muestra una distribución de muestras dentro de los grupos.

Después de entrenar los 5 modelos de DBSCAN con diferentes conjuntos de datos, se compararon las métricas resultantes (tabla 10) y se seleccionó el mejor modelo:

Tabla 10

Tabla de las métricas de calidad de los modelos DBSCAN

	DBSCAN 1	DBSCAN 1_2	DBSCAN 2	DBSCAN 3	DBSCAN 4
Silhouette Score	0.28	-0.04	0.46	0.29	0.59
Calinski Harabasz Score	2.44	2.47	1.51	2.31	1.11
Davies Bouldin Score	373.54	168.66	1956.43	299.99	2987.18

Conclusión: Observando los resultados de los diferentes modelos entrenados, se puede concluir que el modelo dbscan4 (DBSCAN con PCA=7 - datos normalizados) presentó mejores métricas en términos de agrupación, dispersión y definición de los clústeres. Además, garantiza una cantidad adecuada de clústeres, lo cual está alineado con el objetivo final del agrupamiento.

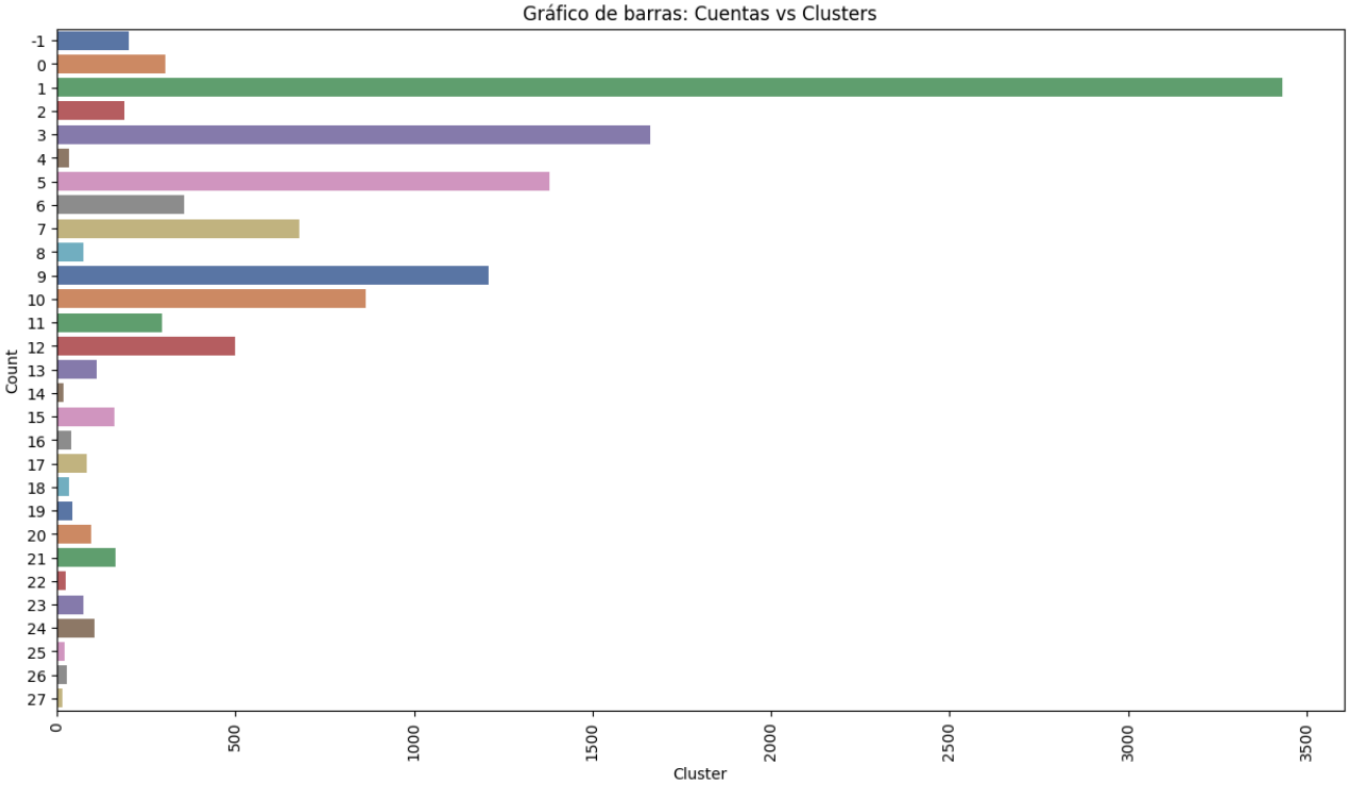
Modelo seleccionado:

```
dbscan4 = DBSCAN(eps=0.4, min_samples=14)
clusters = dbscan4.fit_predict(principal_components7)
```

riesgo: solo se explica el 90% de la variabilidad de los datos.

Figura 50

Gráfico de barras de la cantidad de muestras en cada cluster



(En el grafico anterior observados la cantidad de muestras que hay en cada clúster)

5.3. Aplicación HIERARCHICAL CLUSTERING

Existen dos alternativas para este algoritmo de agrupamiento. En este caso, nos concentramos en el aglomerativo. Se parte del hecho de que cada muestra forma un grupo, por lo que inicialmente existen tantos grupos como muestras. En cada paso del entrenamiento, se van fusionando los grupos más cercanos hasta que queden k grupos o un único grupo. En resumen, comienza con los puntos como grupos individuales y en cada paso combina el par de grupos más cercanos.

Algoritmo básico:

- Calcular la matriz de proximidad.
- Repetir: Fusionar los dos grupos más cercanos. Actualizar la matriz de proximidad para reflejar la proximidad entre el nuevo grupo y los grupos originales.
- Repetir hasta que solo quede un grupo.

Al igual que con el algoritmo de k-medias, la elección del número de grupos debe hacerse cuidadosamente.

- **Paso 1:** Definición del hiperparametro n_cluster. Para determinar el número óptimo de clústeres, se empleó la puntuación de silueta, eligiendo el número de clústeres que maximizara este puntaje:

Figura 51

Método de la silueta datos normalizados

Figura 52

Método de la silueta datos estandarizados

Evaluamos el coeficiente de silueta para los diferentes modelos entrenados con datos normalizados como se puede ver en la figura 51, variando el número de clústeres de 10 a 40. Observamos que los modelos con n_cluster=12 y n_cluster=25 mostraron los mejores puntajes de silueta, por lo que estos fueron considerados en los siguientes pasos. En contraste, los coeficientes de silueta para los modelos entrenados con datos estandarizados (figura 52) son muy bajos cuando el número de clústeres es mayor a 4, por lo que decidimos continuar explorando la técnica solo con los datos normalizados.

5.3.1. Agglomerative Clustering con datos normalizados

- **Paso 2:** Definición Hiperparámetro metric. El hiperparámetro metric determina en el modelo de clustering jerárquico aglomerativo la métrica de distancia que se utilizará para medir la distancia entre los puntos. La elección de la métrica depende de la naturaleza de los datos y del problema que estés tratando de resolver. Para nuestro caso, consideraremos las siguientes métricas:

    - **Euclidiana ('euclidean'):** Es la métrica de distancia más común y se utiliza cuando las dimensiones tienen una escala similar. Mide la longitud del segmento de línea recta que une dos puntos en un espacio euclidiano.

    - **Correlación ('correlation'):** Mide la correlación entre dos variables. Es útil cuando se desea que la distancia refleje la similitud en la forma de los datos, independientemente de su magnitud.

    - **Coseno ('cosine'):** Distancia coseno, que mide el coseno del ángulo entre dos vectores. Se utiliza comúnmente para datos de texto o en espacios de características de alta dimensión.

- **Paso 3:** Definición Hiperparametro Linkaje. El criterio de vinculación determina qué distancia utilizar entre conjuntos de observación. El algoritmo fusionará los pares de clústeres que minimicen este criterio. La elección de este criterio debe tener en cuenta la naturaleza de los datos y el objetivo del problema. Se consideraron los siguientes criterios de vinculación:

    - **Enlace promedio ('average'):** Calcula la distancia media entre todos los pares de puntos en los dos clústeres. Es una buena opción si se desea un equilibrio entre los efectos de los valores atípicos y la estructura global de los datos.

    - **Enlace de Ward ('ward'):** Supone que un conglomerado está representado por su centroide y busca fusionar los grupos que minimicen el incremento en la varianza total dentro de los grupos resultantes. Este método minimiza la varianza al fusionar clústeres, lo que tiende a producir clústeres de tamaño similar. Es menos sensible a los efectos de los valores atípicos en comparación con otros métodos de enlace.

- **Paso 4:** Optimización de hiperparámetros. Se realizó una optimización de los hiperparámetros n_clusters, metric y linkage, y se utilizó el silhouette_score para

evaluar la calidad de los clústeres con el método de optimización GridSearchCV.

```
# Definir el modelo de clustering
model = AgglomerativeClustering()

# Definir el espacio de búsqueda de hiperparámetros
param_grid = {
    'n_clusters': [12, 25],
    'linkage': ['ward', 'average'],
    'metric': ['euclidean', 'correlation','cosine'] }
```

Resultado:

```
Mejores hiperparámetros: {'linkage': 'ward', 'metric': 'euclidean', 'n_clusters': 12}
Mejor puntuación de silueta: 0.4927438960198128
```

- **Paso 5:** Entrenamiento del modelo con los mejores hiperparametros.

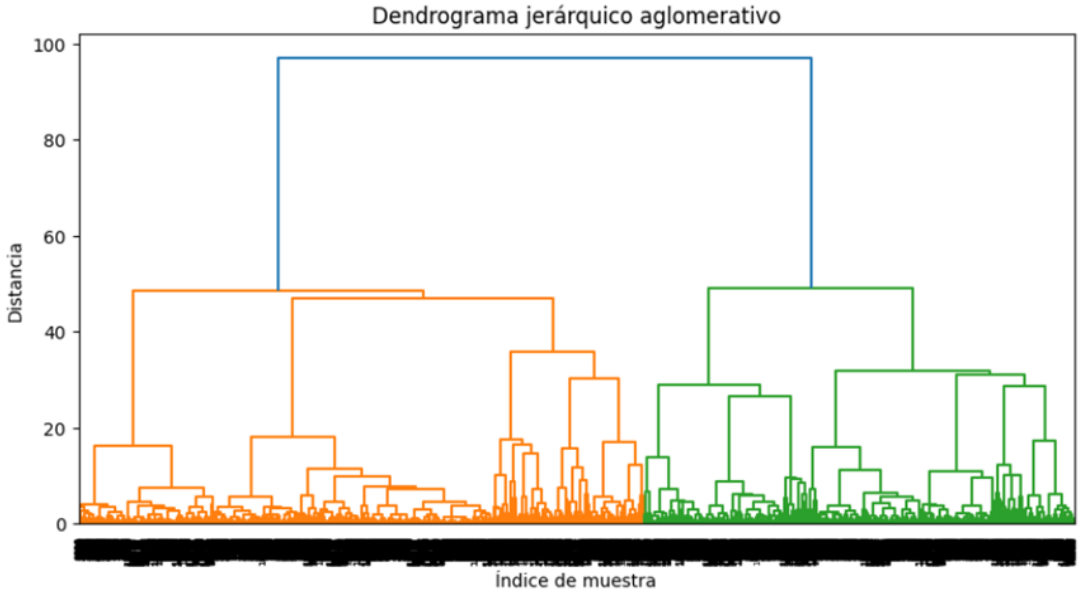
```
# Entrenamiento del modelo conm los mejores hiperparametros
model_HC1 = AgglomerativeClustering(n_clusters=12, metric='euclidean', linkage='ward')

# Entrenar el modelo
    model_HC1.fit(scaled_data)
```

Las métricas resultantes:

```
Coeficiente de silueta: 0.4011491641781883
Índice de Calinski-Harabasz: 2910.19180019344
Puntuación de Davies-Bouldin: 1.2050866573201404
```

Figura 53

Dendograma clustering jerárquico datos estandarizados



(Calcula las fusiones de clusters utilizando el método de enlace de Ward y la distancia euclidiana)

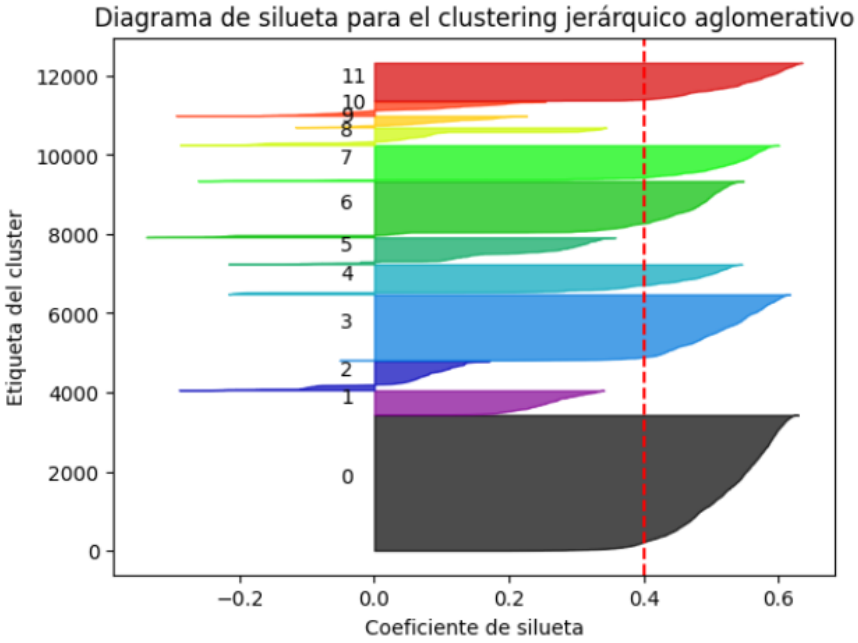
Se calculó el coeficiente de correlación cophenético con el fin de medir la calidad de un agrupamiento jerárquico. Los valores cercanos a 1 indican que el agrupamiento conserva bien las relaciones de distancia originales entre los puntos, sugiriendo una alta calidad en la agrupación.

c, coph_dists = cophenet(Z, pdist(scaled_data))

resultado: 0.7026101783149373

Figura 54

Diagrama de silueta clustering jerárquico datos estandarizados



Como resultado del entrenamiento del modelo de clustering jerárquico aglomerativo con 12 grupos como se observa en la figura 54, utilizando la métrica de distancia euclidiana, el criterio de vinculación de Ward y los datos normalizados, se observó que los grupos no son homogéneos y que algunas muestras están clasificadas de manera errónea. Por lo tanto, se repitió el proceso anterior reduciendo la dimensionalidad de los datos, en este caso utilizando las 7 componentes principales que retienen el 90% de la variabilidad de los datos.

5.3.2. Agglomerative Clustering con datos normalizados PCA=7

- **Paso 1:** Definición del hiperparametro n_cluster.

Figura 55

Método de la silueta datos normalizados

Observando la figura 55 se seleccionó los valores para el numero de clústeres de 22 y 28 ya que para estos se presenta un alto score de silueta**.**

- Para los pasos 2 y 3 se tendrán en cuenta los mismos valores de los hiperparametros (Metric y Linkaje)

- **Paso 4:** Optimización de hiperparametros.

```
# Definir el modelo de clustering
model = AgglomerativeClustering()

# Definir el espacio de búsqueda de hiperparámetros
param_grid = {
    'n_clusters': [22, 28],
    'linkage': ['ward', 'average'],
    'metric': ['euclidean', 'correlation','cosine'] }
```

Resultado:

```
Mejores hiperparámetros: {'linkage': 'ward', 'metric': 'euclidean', 'n_clusters': 22}
Mejor puntuación de silueta: 0.7867079498870482
```

- **Paso 5:** Entrenamiento del modelo con los mejores hiperparametros.

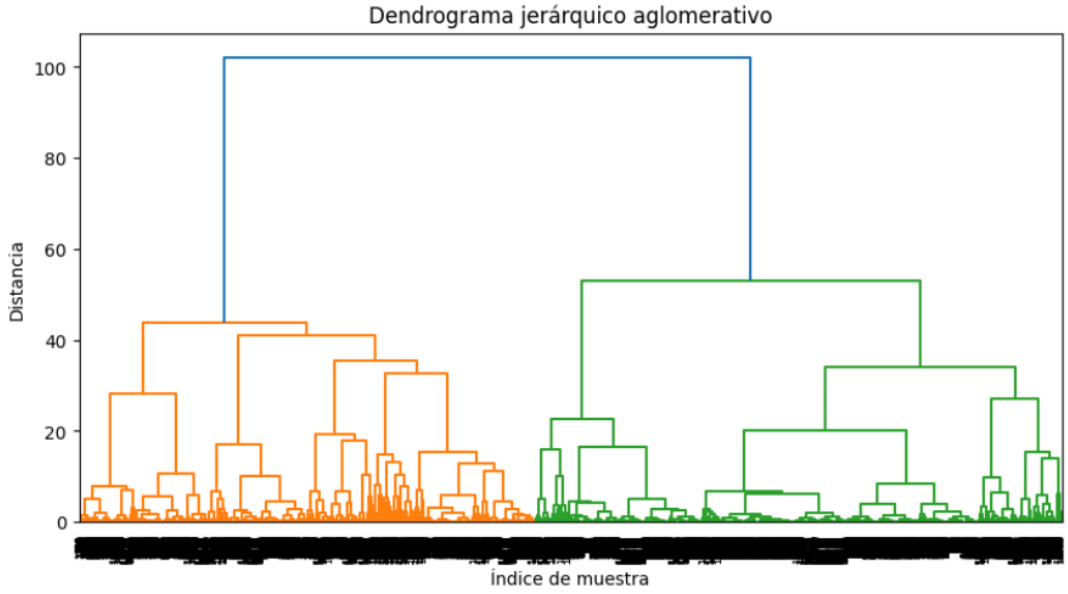
```
# Entrenamiento del modelo conm los mejores hiperparametros
model_HC2 = AgglomerativeClustering(n_clusters=22, metric='euclidean', linkage='ward')

# Entrenar el modelo
      model_HC2.fit(principal_components7)
```

Las métricas resultantes:

Coeficiente de silueta: 0.5499755845768477
Índice de Calinski-Harabasz: 4843.838514164271
Puntuación de Davies-Bouldin: 0.8843356666486567

Figura 56

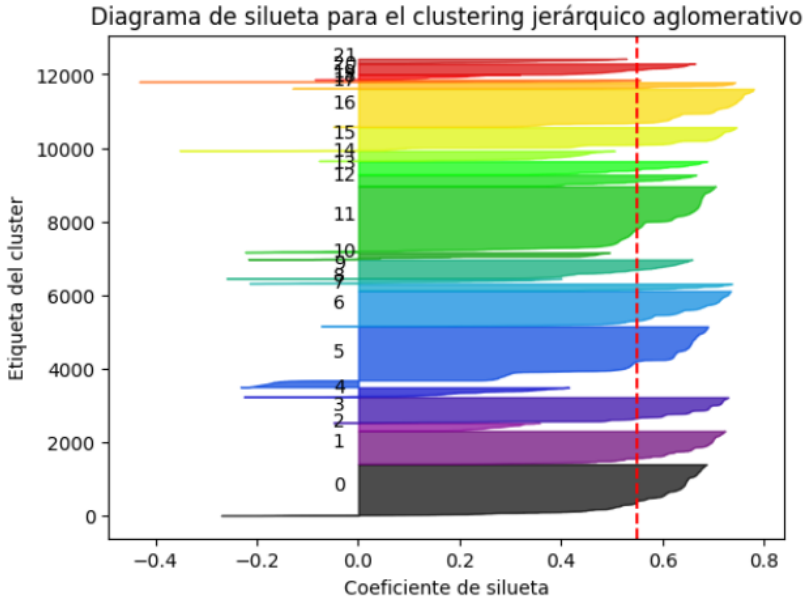
Dendograma clustering jerárquico datos normalizados



c, coph_dists = cophenet(Z, pdist(principal_components7))
Resultado: 0.7484132447061911

Figura 57

Diagrama de silueta clustering jerárquico datos normalizados



Se analizó el diagrama de silueta de la figura 57, evidenciando que los grupos no son homogéneos en cuanto a la cantidad de muestras y que algunas muestras fueron clasificadas incorrectamente.

Después de haber entrenado los dos modelos con los diferentes conjuntos de datos, se compararon las diversas métricas resultantes (ver tabla 11), con base en estos se seleccionó el mejor modelo:

Tabla 11

Tabla de las métricas de calidad de los modelos Agglomerative Clustering

	HC1	HC2
Silhouette Score	0.4	0.55
Calinski Harabasz Score	1.21	0.88
Davies Bouldin Score	2910.19	4843.84

De acuerdo con los dos modelos entrenados, observamos que los resultados son moderadamente buenos al utilizar la normalización como técnica de transformación de datos. Como se esperaba, obtuvimos mejores resultados al reducir la dimensionalidad de los datos, lo que nos permitió obtener métricas mejoradas y una mayor cantidad de clústeres, lo cual es ideal para nuestro proyecto.

Conclusión: De los dos modelos entrenados con la técnica de Agglomerative Clustering, se seleccionó HC2, el cual fue aplicado con 7 componentes y un número de clústeres de 22, utilizando la métrica de distancia euclidiana y el criterio de vinculación de Ward.

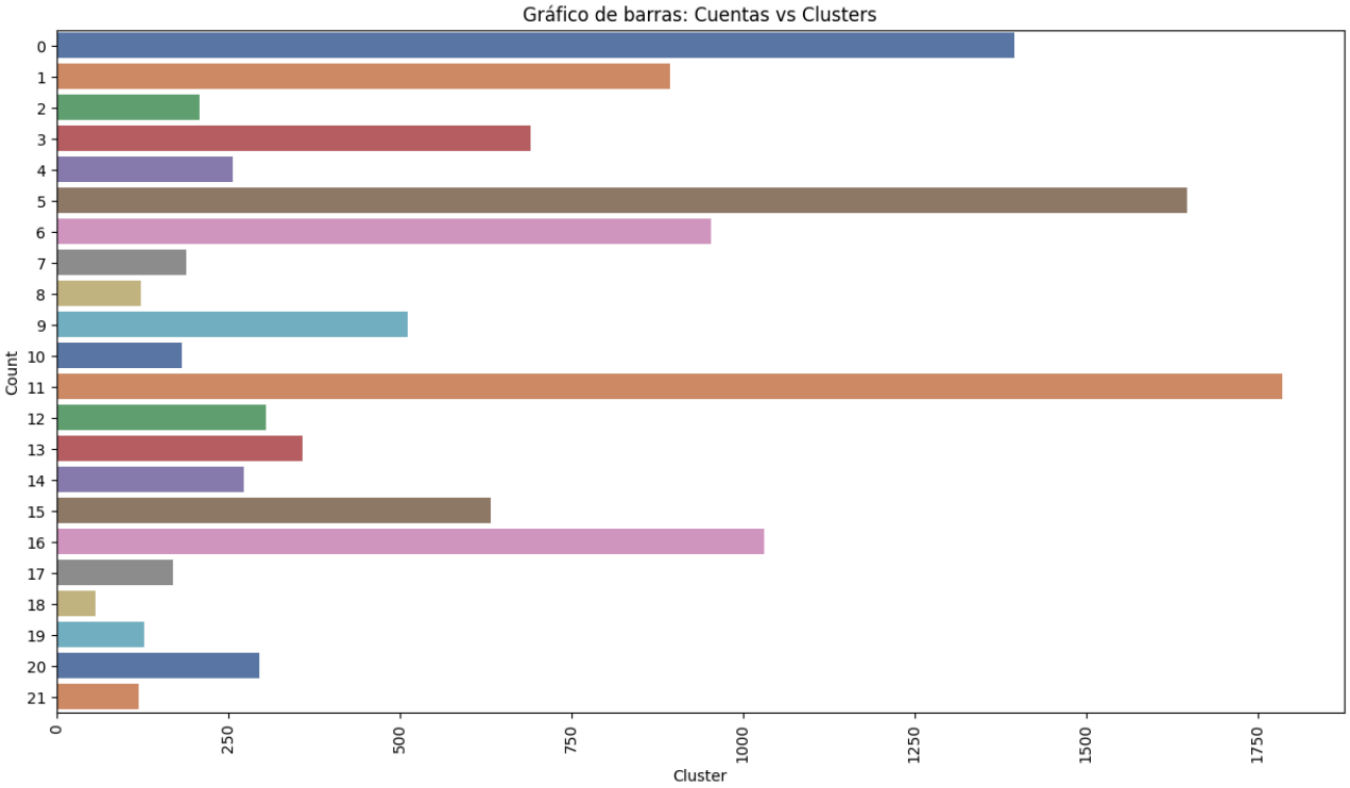
Modelo seleccionado:

```
model_HC2 = AgglomerativeClustering(n_clusters=22, metric='euclidean', linkage='ward')
model_HC2.fit(principal_components7)
labels_HC2 = model_HC2.labels_
```

Riesgo: solo se explica el 90% de la variabilidad de los datos.

Figura 58

Gráfico de barras de la cantidad de muestras en cada cluster



(En el grafico anterior observados la cantidad de muestras que hay en cada clúster)

5.4. Aplicación AFFINITY PROPAGATION

Affinity Propagation fue seleccionado por las siguientes propiedades:

- Puede determinar automáticamente el número óptimo de clústeres basado en los datos, sin necesidad de ajustes manuales.
- Puede generar agrupaciones de alta calidad que sean coherentes y diversas, ya que selecciona ejemplares que son representativos y distintos entre sí.
- Es ideal para conjuntos de datos grandes.

El algoritmo cuenta con algunos parámetros relevantes que tienen una gran influencia en la formación de los clústeres:

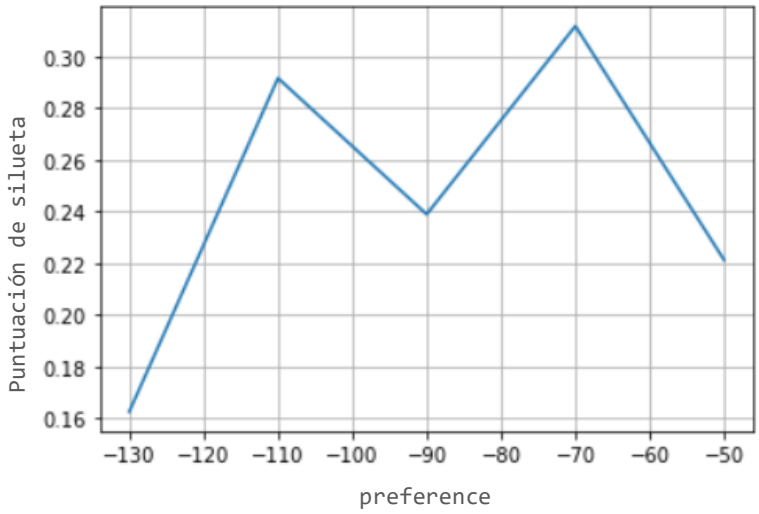
- ***Damping*** **(factor de amortiguación):** Este parámetro existe para la estabilización numérica y puede considerarse como una tasa de aprendizaje que converge lentamente. Los autores recomiendan elegir un factor de amortiguación dentro del rango de 0.5 a 1. El valor predeterminado es 0.5.***max_iter*** **(número máximo de iteraciones):** el valor predeterminado es 200.

- ***preference* (preferencia):** representa la entrada que indica la probabilidad de que un punto de datos sea elegido como ejemplo. Influye en la cantidad de grupos que se formarán y en qué puntos de datos se asignarán como ejemplos. Los puntos con valores de preferencia más altos tienen más probabilidades de ser elegidos ejemplos. Este parámetro no tiene un valor predeterminado establecido.

Era necesario ajustar estos parámetros para obtener agrupaciones de la mayor calidad posible. Inicialmente, se ajustó el parámetro *preference* (ver figura 59). Se probaron varios valores y se compararon las puntuaciones de silueta para determinar el valor óptimo.

Figura 59

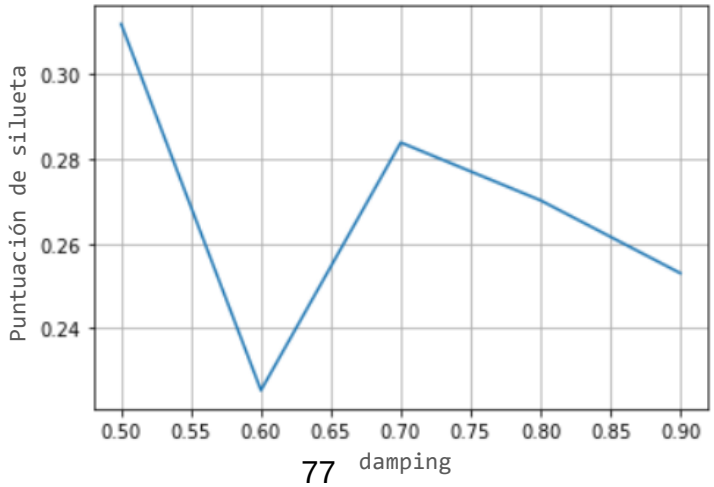
Gráfico de las puntaciones de silueta para seleccionar el parámetro preference



En este caso, el valor óptimo para el parámetro de preferencia fue -70, ya que obtuvo la puntuación más alta de silueta, superior a 0.30. Para seleccionar el parámetro *damping*, se siguió un procedimiento similar. Se generó el mismo gráfico manteniendo fijo el parámetro *preference* en -70.

Figura 60

Gráfico de las puntaciones de silueta para seleccionar el parámetro damping



En este caso como se ve en la figura 60, el valor óptimo para el factor de amortiguación fue 0.5, ya que obtuvo la puntuación más alta de silueta, superior a 0.30.

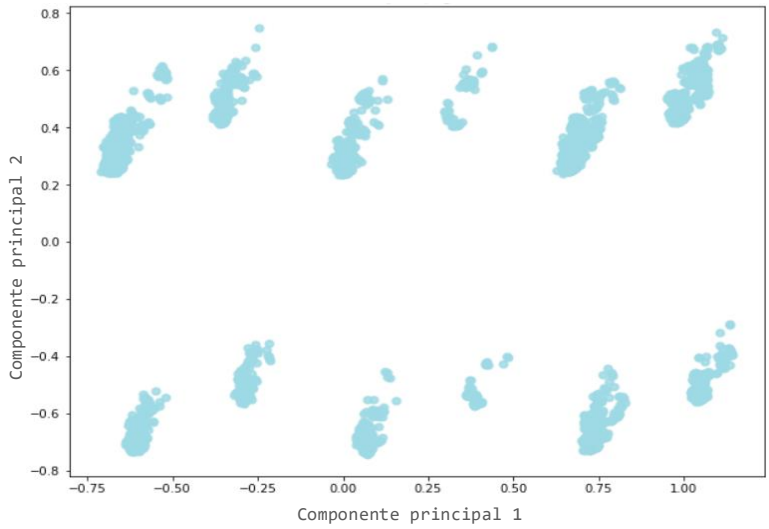
Por otro lado, al construir los gráficos se observó que el algoritmo tardaba en ejecutar 200 iteraciones, por lo tanto, se redujo el número de iteraciones a 100 para todos los modelos, excepto el último.

5.4.1. Modelo 1: Affinity Propagation con PCA=10

- **Paso 1:** Como se observó anteriormente en los resultados del PCA, con 10 componentes se explica al menos el 95% de la varianza total de los datos. Por tanto, se entrenó un modelo con estas componentes. Bajo este modelo se conformaron 1,961 agrupaciones y se realizó un gráfico con las 2 primeras componentes del PCA para observar los clústeres (ver figura 61).

Figura 61

Gráfico de los grupos formados por el modelo Affinity Propagation con PCA=10



Desde el punto de vista grafico no son claras las agrupaciones que formó el algoritmo.

- **Paso 2:** También se generaron las métricas de silueta, Calinski-Harabasz y Davies-Bouldin para evaluar la calidad de las agrupaciones.

```
Silhouette Score: 0.21569571741292695
Calinski Harabasz Score: 57.61073037929948
Davies Bouldin Score: 0.882754568795795
```

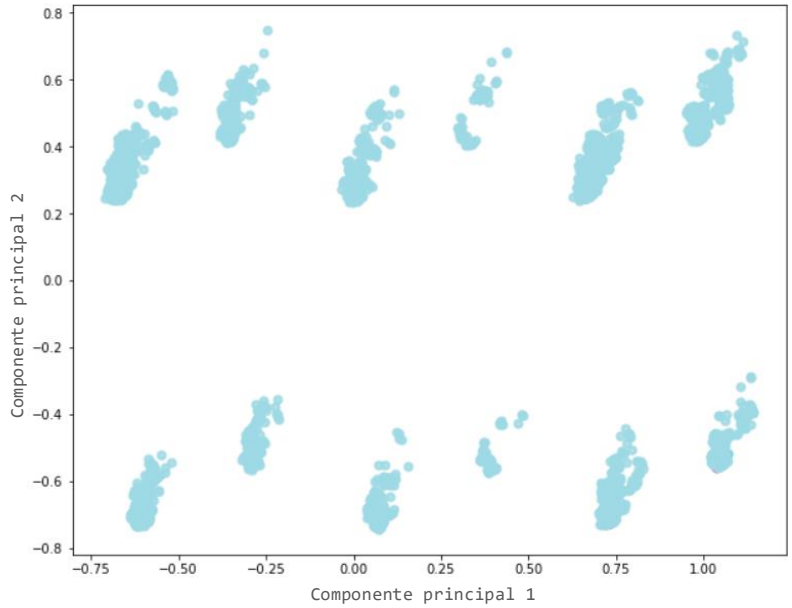
5.4.2. Modelo 2: Affinity Propagation con PCA=15

- **Paso 1:** Buscando mejorar los resultados obtenidos con PCA, se ajustó nuevamente un modelo con esta técnica de reducción de dimensionalidad. Se

tomaron 15 componentes, pues con estas se explicaba el 99% de la variabilidad de los datos. Bajo este modelo se conformaron 2,077 agrupaciones y se realizó un gráfico con las 2 primeras componentes del PCA para observar los clústeres.

Figura 62

Gráfico de los grupos formados por el modelo Affinity Propagation con PCA=15



Desde el punto de vista grafico no son claras las agrupaciones que formó el algoritmo (figura 62).

- **Paso 2:** También se generaron las métricas de silueta, Calinski-Harabasz y Davies-Bouldin para evaluar la calidad de las agrupaciones.

```
Silhouette Score: 0.11340177796878488
Calinski Harabasz Score: 39.37513453347376
Davies Bouldin Score: 0.8601643361392253
```

La puntuación de silueta y Calinski-Harabasz empeoró para este modelo; sin embargo, la métrica Davies-Bouldin mejoró, ya que cuanto más cercano esté a 0, mejor será la separación y la calidad de las agrupaciones.

5.4.3. Modelo 3: Affinity Propagation con datos normalizados

- **Paso 1:** Después de observar que los resultados con PCA no fueron los esperados, optamos por ajustar el modelo utilizando los datos normalizados. En este caso, la cantidad de clústeres se redujo considerablemente, obteniendo únicamente 35 clústeres.

- **Paso 2:** También se generaron las métricas de silueta, Calinski-Harabasz y Davies-Bouldin para evaluar la calidad de las agrupaciones.

```
Silhouette Score: 0.31184048425884425
Calinski Harabasz Score: 1583.600391689316
Davies Bouldin Score: 1.9397879744416218
```

En este modelo, las métricas de silueta y Calinski-Harabasz mejoraron considerablemente, especialmente esta última. Sin embargo, la métrica Davies-Bouldin mostró un peor desempeño.

5.4.4. Modelo 4: Affinity Propagation con datos estandarizados

- **Paso 1:** Posteriormente al ajuste del modelo con los datos normalizados, se realizó el mismo ejercicio pero empleando los datos estandarizados sin ninguna técnica de reducción de dimensionalidad. En este caso, se obtuvieron 373 clústeres, una cantidad mayor que en el modelo 3 pero menor que en los modelos 1 y 2.

- **Paso 2:** También se generaron las métricas de silueta, Calinski-Harabasz y Davies-Bouldin para evaluar la calidad de las agrupaciones.

```
Silhouette Score: 0.15076342086015468
Calinski Harabasz Score: 209.0533329738467
Davies Bouldin Score: 1.4757756313892871
```

En este caso, el puntaje de silueta fue mayor que en el modelo 2 pero menor que en los modelos 1 y 3. En cuanto a la métrica de Calinski-Harabasz, solo fue superada por el modelo 3.

5.4.5. Modelo 5: Affinity Porpagation con daos normalizados y 200 iteraciones

- **Paso 1:** Teniendo en cuenta que el mejor modelo fue el ajustado con los datos normalizados, se decidió crear un último modelo aumentando el número de iteraciones de 100 a 200, con el fin de mejorar las métricas de calidad de las agrupaciones. En este caso, se obtuvo la menor cantidad de clústeres en comparación con los otros modelos, con un total de 24 agrupaciones realizadas por el algoritmo.

- **Paso 2:** También se generaron las métricas de silueta, Calinski-Harabasz y Davies-Bouldin para evaluar la calidad de las agrupaciones.

```
Silhouette Score: 0.3285624422389157
Calinski Harabasz Score: 2210.6112365161925
Davies Bouldin Score: 1.299239440405837
```

Al aumentar el número de iteraciones, se obtuvo el resultado esperado, ya que las métricas del modelo 5 mejoraron en comparación con las métricas obtenidas con el modelo 3.

Conclusión: Finalmente, de los modelos ajustados con esta métrica, el que mostró mejor rendimiento fue el modelo 5 con los datos normalizados y 200 iteraciones.

Modelo seleccionado:

```
af_5 = AffinityPropagation(damping= 0.5, preference=-70, affinity = 'euclidean', max_iter=200,
    random_state=14).fit(scaled_data)
cluster_centers_indices_5 = af_5.cluster_centers_indices_
labels_5 = af_5.labels_
```

6. VALIDAR TÉCNICAS ESTADÍSTICAS PARA EVALUAR LA HOMOGENEIDAD DE LOS GRUPOS FORMADOS POR LOS DIFERENTES MODELOS ML, PARA IDENTIFICAR EL MODELO MAS ADECUADO ENTRE LOS ALGORITMOS ENTRENADOS.

Con el fin de validar la calidad de los grupos formados por los algoritmos, se optó por emplear una técnica estadística. La idea era encontrar una métrica que permitiera determinar si los clústeres son heterogéneos entre sí. Esto indicaría que las características de los clientes dentro de cada grupo son similares (y diferentes de las de otros grupos), lo que implica que cada clúster sería homogéneo internamente.

Entre las técnicas estadísticas consideradas para este análisis, se buscó una prueba que cumpliera con las siguientes características:

1. **No paramétrica:** Los datos no necesitan cumplir supuestos de normalidad.
2. **Multivariante:** Dado que las observaciones están medidas en varias variables.
3. **Métrica de referencia:** La prueba debería calcular una métrica objetiva como el Valor-p para una decisión estadística.

Por estas razones, se seleccionó la prueba de permutación multivariada, una prueba no paramétrica que contrasta dos hipótesis y permite tomar decisiones basadas en el Valor-p. Esta prueba es robusta, lo que significa que puede manejar grupos desiguales y datos atípicos.

De cada técnica de modelado se seleccionó el algoritmo que mostró el mejor desempeño:

- **K-Means:** PCA=5 con datos estandarizados.
- **DBSCAN:** PCA=7 con datos normalizados.
- **Agglomerative Clustering:** PCA=7, distancia euclidiana y criterio de vinculación Ward.
- **Affinity Propagation:** Datos normalizados con Affinity Propagation.

Se aplicó la prueba a cada uno de los clústeres formados por los algoritmos y se obtuvo el Valor-p de cada técnica. Como estadístico de prueba se eligió la mediana, una métrica robusta frente a datos atípicos en comparación con la media.

Las hipótesis contrastadas fueron las siguientes:

H_0: No hay diferencias significativas entre las medianas de los grupos
H_1: Hay diferencias significativas entre las medianas de los grupos

Los resultados fueron los siguientes:

Tabla 12

Tabla del Valor-p obtenido para cada una de las técnicas

Técnica	Cantidad de clusters	Valor-p
K-Means	39	0.784
DBSCAN	27	0.739
Agglomerative Clustering	21	0.770
Affinity Porpagation	23	0.759

El Valor-p fue similar en todos los casos.

Los resultados expuestos en la tabla 12 indicaron que no había evidencia suficiente para rechazar la hipótesis nula, es decir, no se encontró suficiente evidencia para afirmar que las medianas por clúster eran significativamente diferentes. Por lo tanto, según este Valor-p, no se encontró suficiente evidencia para concluir que existían diferencias significativas entre los clústeres formados por cada una de las técnicas. Esto sugiere que los grupos no son heterogéneos entre sí y que los clústeres no tienen una estructura homogénea internamente.

Estos resultados podrían implicar que sería beneficioso replicar el modelado de los datos incluyendo más observaciones, empleando técnicas de muestreo, modificando variables existentes o incorporando nuevas variables a los datos. Además, sería útil explorar otras técnicas de clustering.

7. RECONOCER CARACTERISTICAS Y PATRONES EN LOS GRUPOS DE CLIENTES RESULTANTES DE LOS MODELOS IMPLEMENTADOS, QUE PUEDAN SER RELEVANTES PARA DETECTAR POSIBLES OPERACIONES SOSPECHOSAS.

Como se mencionó anteriormente, la técnica que produjo los mejores grupos fue K-means con 5 componentes. Por lo tanto, se creó un tablero con los resultados de esta técnica para facilitar la interacción y el análisis de las características de los clústeres formados. Este tablero incluye visualizaciones generales de todos los clústeres, así como visualizaciones específicas para cada uno de ellos, lo que facilita el reconocimiento y el análisis detallado de sus diferentes atributos.

Figura 63

Variables de riesgo visualización general de todos los clústeres

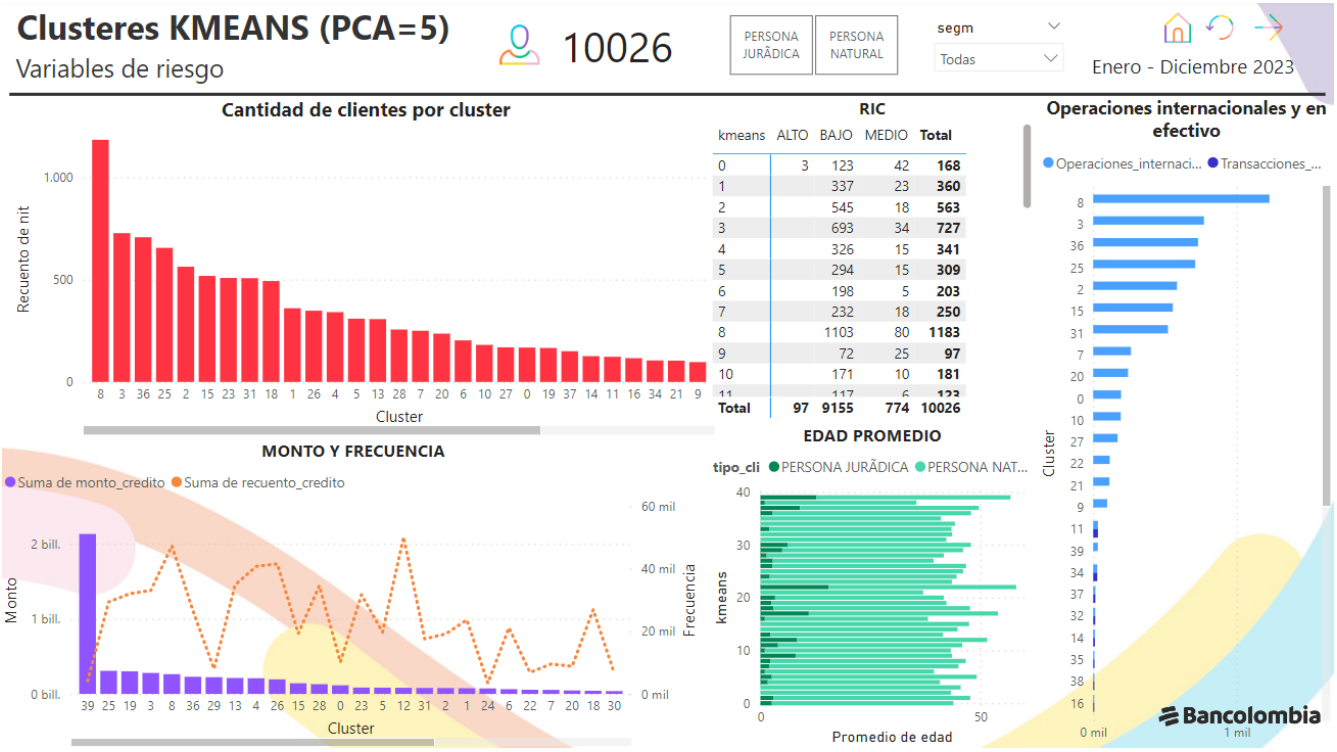
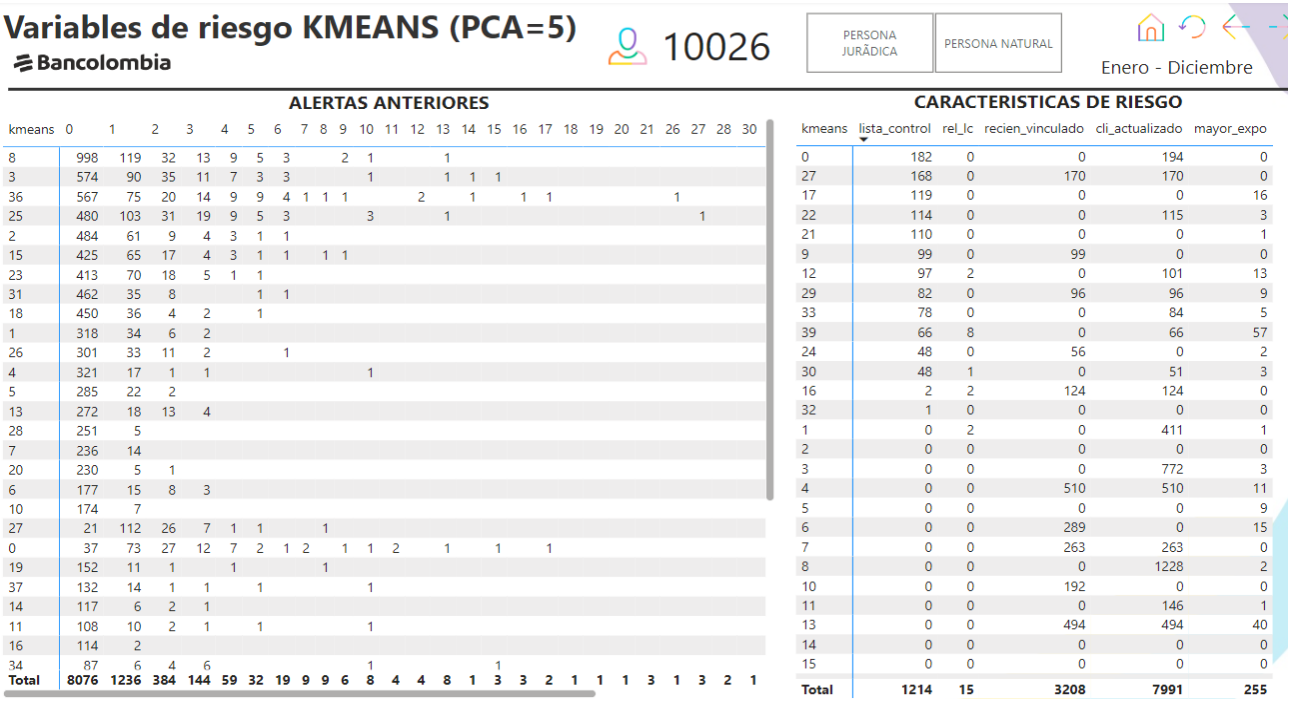


Figura 64

Variables de riesgo visualización general de todos los clústeres



ALERTAS ANTERIORES

kmeans	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	26	27	28	30
8	998	119	32	13	9	5	3			2	1			1												
3	574	90	35	11	7	3	3				1			1	1	1										
36	567	75	20	14	9	9	4	1	1	1			2		1		1	1					1			
25	480	103	31	19	9	5	3				3			1										1		
2	484	61	9	4	3	1	1																			
15	425	65	17	4	3	1	1		1	1																
23	413	70	18	5	1	1																				
31	462	35	8			1	1																			
18	450	36	4	2		1																				
1	318	34	6	2																						
26	301	33	11	2			1																			
4	321	17	1	1							1															
5	285	22	2																							
13	272	18	13	4																						
28	251	5																								
7	236	14																								
20	230	5	1																							
6	177	15	8	3																						
10	174	7																								
27	21	112	26	7	1	1				1																
0	37	73	27	12	7	2	1	2		1	1	2		1		1		1								
19	152	11	1		1				1																	
37	132	14	1	1		1					1															
14	117	6	2	1																						
11	108	10	2	1		1					1															
16	114	2																								
34	87	6	4	6							1						1									
Total	**8076**	**1236**	**384**	**144**	**59**	**32**	**19**	**9**	**9**	**6**	**8**	**4**	**4**	**8**	**1**	**3**	**3**	**2**	**1**	**1**	**1**	**3**	**1**	**3**	**2**	**1**

CARACTERISTICAS DE RIESGO

kmeans	lista_control	rel_lc	recien_vinculado	cli_actualizado	mayor_expo
0	182	0	0	194	0
27	168	0	170	170	0
17	119	0	0	0	16
22	114	0	0	115	3
21	110	0	0	0	1
9	99	0	99	0	0
12	97	2	0	101	13
29	82	0	96	96	9
33	78	0	0	84	5
39	66	8	0	66	57
24	48	0	56	0	2
30	48	1	0	51	3
16	2	2	124	124	0
32	1	0	0	0	0
1	0	2	0	411	1
2	0	0	0	0	0
3	0	0	0	772	3
4	0	0	510	510	11
5	0	0	0	0	9
6	0	0	289	0	15
7	0	0	263	263	0
8	0	0	0	1228	2
10	0	0	192	0	0
11	0	0	0	146	1
13	0	0	494	494	40
14	0	0	0	0	0
15	0	0	0	0	0
Total	**1214**	**15**	**3208**	**7991**	**255**

Figura 65

Características generales de todos los clústeres

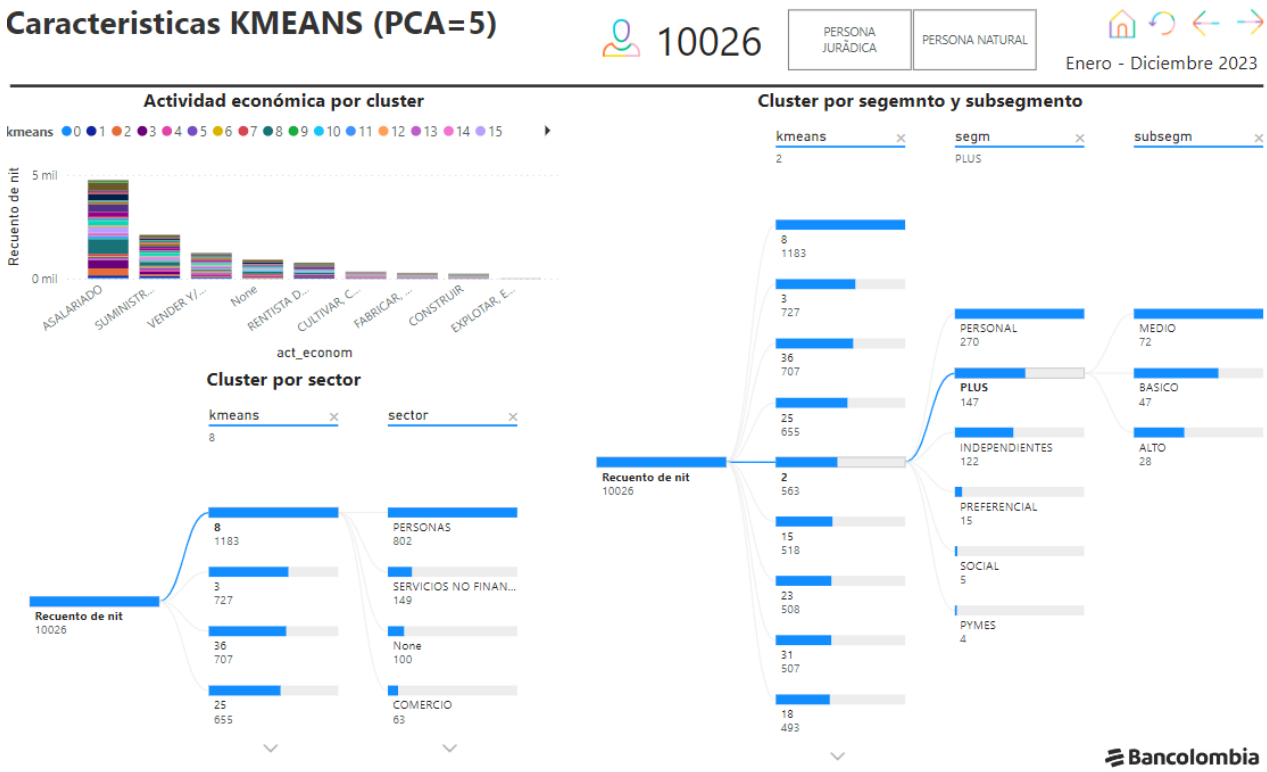
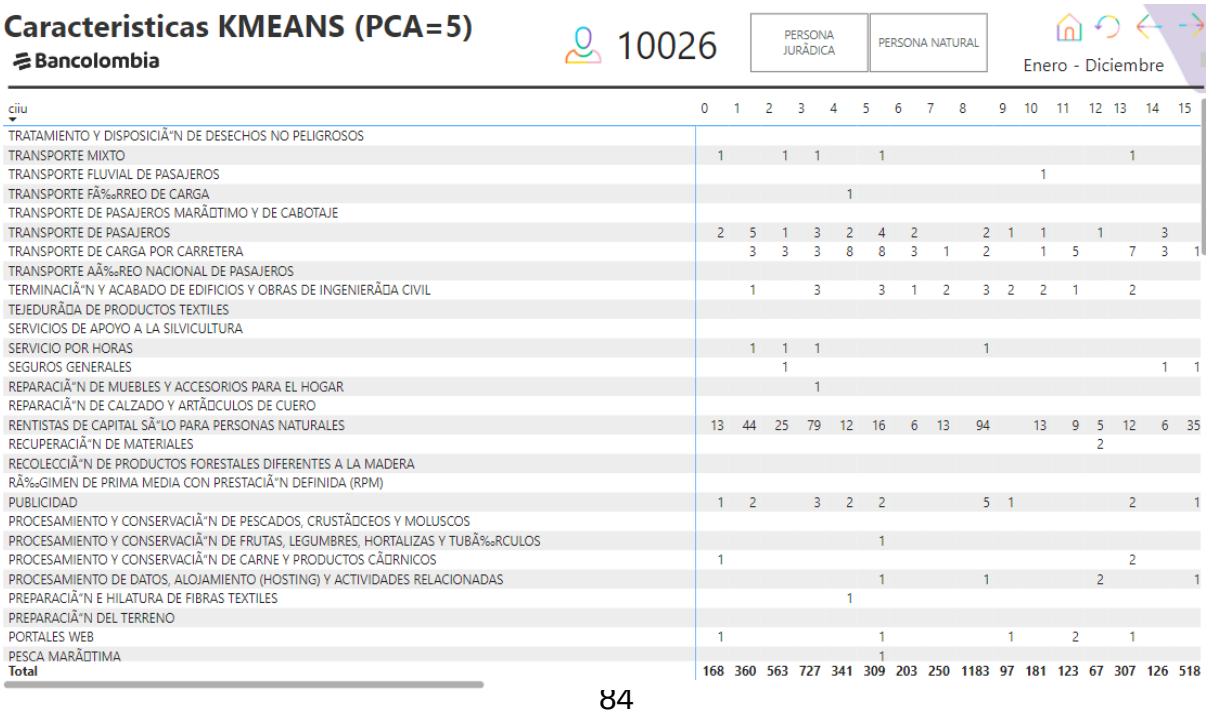


Figura 66

CIIU de todos los clústeres



ciiu	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
TRATAMIENTO Y DISPOSICIÃ"N DE DESECHOS NO PELIGROSOS																
TRANSPORTE MIXTO	1		1	1		1								1		
TRANSPORTE FLUVIAL DE PASAJEROS											1					
TRANSPORTE FÃ‰RREO DE CARGA					1											
TRANSPORTE DE PASAJEROS MARÃ□TIMO Y DE CABOTAJE																
TRANSPORTE DE PASAJEROS	2	5	1	3	2	4	2		2	1	1		1		3	
TRANSPORTE DE CARGA POR CARRETERA		3	3	3	8	8	3	1	2		1	5		7	3	1
TRANSPORTE AÃ‰REO NACIONAL DE PASAJEROS																
TERMINACIÃ"N Y ACABADO DE EDIFICIOS Y OBRAS DE INGENIERÃ□A CIVIL		1		3		3	1	2	3	2	2	1		2		
TEJEDURÃ□A DE PRODUCTOS TEXTILES																
SERVICIOS DE APOYO A LA SILVICULTURA																
SERVICIO POR HORAS		1	1	1					1							
SEGUROS GENERALES			1												1	1
REPARACIÃ"N DE MUEBLES Y ACCESORIOS PARA EL HOGAR				1												
REPARACIÃ"N DE CALZADO Y ARTÃ□CULOS DE CUERO																
RENTISTAS DE CAPITAL SÃ"LO PARA PERSONAS NATURALES	13	44	25	79	12	16	6	13	94		13	9	5	12	6	35
RECUPERACIÃ"N DE MATERIALES													2			
RECOLECCIÃ"N DE PRODUCTOS FORESTALES DIFERENTES A LA MADERA																
RÃ‰GIMEN DE PRIMA MEDIA CON PRESTACIÃ"N DEFINIDA (RPM)																
PUBLICIDAD	1	2		3	2	2			5	1				2		1
PROCESAMIENTO Y CONSERVACIÃ"N DE PESCADOS, CRUSTÃ□CEOS Y MOLUSCOS																
PROCESAMIENTO Y CONSERVACIÃ"N DE FRUTAS, LEGUMBRES, HORTALIZAS Y TUBÃ‰RCULOS						1										
PROCESAMIENTO Y CONSERVACIÃ"N DE CARNE Y PRODUCTOS CÃ□RNICOS	1													2		
PROCESAMIENTO DE DATOS, ALOJAMIENTO (HOSTING) Y ACTIVIDADES RELACIONADAS						1			1				2			1
PREPARACIÃ"N E HILATURA DE FIBRAS TEXTILES					1											
PREPARACIÃ"N DEL TERRENO																
PORTALES WEB	1					1				1		2		1		
PESCA MARÃ□TIMA						1										
Total	**168**	**360**	**563**	**727**	**341**	**309**	**203**	**250**	**1183**	**97**	**181**	**123**	**67**	**307**	**126**	**518**

Figura 67

Visualización de características de cada clúster

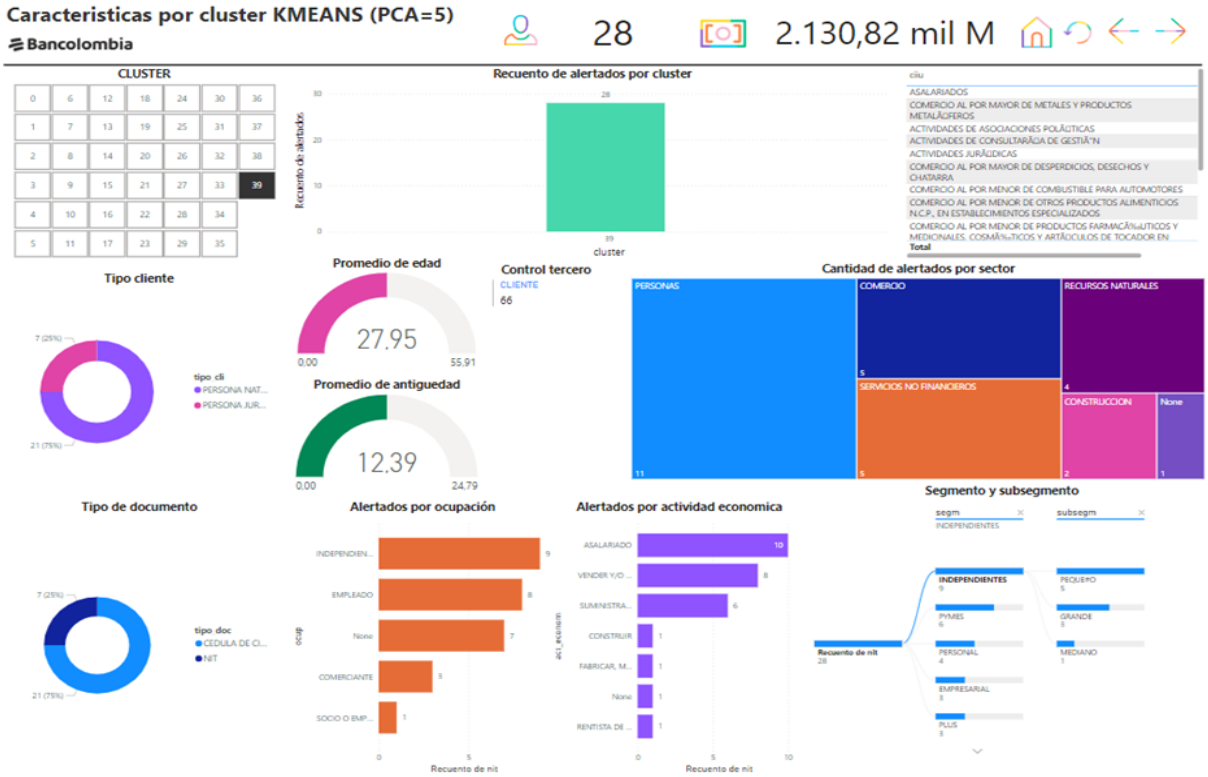
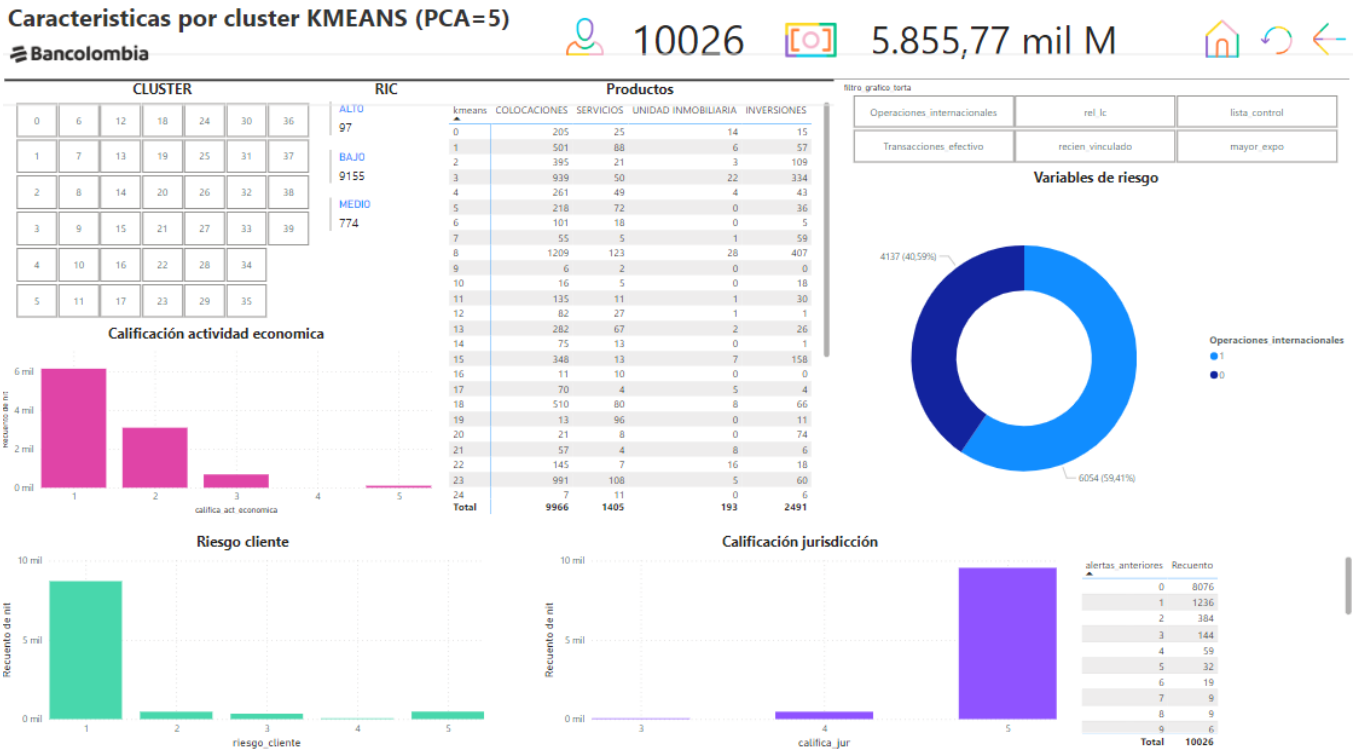


Figura 68

Visualización de características de cada clúster



La exploración del tablero se llevó a cabo con un experto en el área de investigación que está familiarizado con el objetivo del tablero. Comenzamos con las visualizaciones generales (figuras 63, 64, 65 y 66), donde destacó el clúster 39 por registrar el mayor volumen de transacciones durante el período de análisis de 2023, superando los 2 billones de pesos. También llamó la atención el clúster 23, que incluía a 508 individuos, todos con alertas por operaciones en efectivo, uno de los canales más utilizados para el lavado de activos y la financiación del terrorismo debido a la pérdida de trazabilidad de los recursos. Se seleccionaron estos dos clústeres para proceder con el examen detallado de las demás visualizaciones (figuras 67 y 68), identificando las características específicas de cada uno:

- **Clúster 39:** Agrupó a 28 individuos, siendo el clúster con el menor número de personas. Siete de ellos son personas jurídicas, mientras que los demás son personas naturales. Es importante mencionar que en el set de entrenamiento solo hay 206 personas jurídicas disponibles. Veintiséis personas en este clúster tienen un alto RIC (Riesgo de Integridad y Cumplimiento), que es un indicador de riesgo de lavado de activos y financiamiento del terrorismo para los clientes de la entidad. Solo una persona ha sido alertada por primera vez; el resto tiene alertas anteriores, con seis personas que han acumulado más de 13 alertas. El monto promedio transado es de 62 mil millones de pesos, el más alto entre todos los clústeres, lo que indica características de alto riesgo LAFT (Lavado de Activos y Financiamiento del Terrorismo).
- **Clúster 23:** Agrupó a 508 individuos, todas personas naturales, con un promedio de edad de 43 años y una antigüedad promedio en la entidad de 12 años. El 81% de estos clientes han sido alertados por primera vez. Hay una concentración en las actividades económicas de asalariados y proveedores de servicios, así como en el segmento comercial independiente. En términos transaccionales, muestran una alta frecuencia de transacciones con un monto total promedio de 150 millones durante el periodo de análisis de 2023.Podemos ver que los clústeres de estudio cuentan con características de riesgo similares entre si suficientes como para sugerir un buen agrupamiento.

De acuerdo con la información anterior, se concluyó lo siguiente: la exploración de resultados debe siempre contar con la presencia de un experto en investigación. Esto permitió identificar algunos grupos que, estadísticamente, no son los óptimos, considerando el riesgo operativo potencial. Esto resalta la importancia de la colaboración funcional que conecta los resultados con la estrategia del negocio. Además, las alertas se basan en tipologías LAFT, y cada cliente alertado presenta particularidades únicas. Por lo tanto, la percepción del riesgo por parte del investigador guía la investigación; lo que puede ser riesgoso para uno, puede no serlo para otro.

Aunque algunas agrupaciones pueden ser útiles, tratar a muchos clientes de la misma manera puede ser riesgoso debido a sus particularidades individuales que requieren gestiones diferenciadas, definidas por el proceso de investigación. Esto podría pasarse por alto en un enfoque masivo. Por lo tanto, además de evaluar la calidad estadística de los clústeres, es crucial evaluar su calidad desde una perspectiva basada en riesgos, como se hizo con los clústeres 39 y 23, para tomar decisiones informadas sobre si utilizar o descartar los agrupamientos sugeridos por el modelo.

En resumen, aunque los grupos generados por el modelo puedan tener méritos estadísticos, no se recomienda su uso sin una exploración exhaustiva por parte de un experto. Cualquier decisión debe ser respaldada por un análisis funcional que considere aspectos estratégicos y de gestión de riesgos

para garantizar que el grupo esté correctamente definido. Este ejercicio subraya la complejidad de las señales de alerta LAFT y del proceso investigativo, reflejado a lo largo de la historia del área.

8. EMPLEAR TÉCNICAS ANALÍTICAS PARA MEJORAR LA CAPACIDAD DE ANÁLISIS DEL ÁREA DE INVESTIGACIÓN, EN LUGAR DE EMPLEAR MÉTODOS INTUITIVOS QUE REQUIEREN UN MAYOR TIEMPO DE EJECUCIÓN.

Debido al alto volumen de alertas mensuales generadas por los modelos analíticos detectivos, se asignaba la capacidad de un analista para desarrollar estrategias. Estas estrategias implicaban la creación de reglas e intersecciones de datos a nivel experto para agrupar clientes alertados, con el objetivo de mejorar la eficiencia en el cierre de alertas. Sin embargo, estas estrategias eran empíricas, requerían al menos dos días de dedicación y estaban altamente influenciadas por el apetito de riesgo del analista. Carecían de respaldo estadístico o matemático, lo cual se reflejaba en decisiones de cierre que a menudo resultaban en reprocesos debido a la falta de consideración de las particularidades de los clientes y su riesgo potencial, resultando en nuevas alertas que requerían reinvestigación.

Con base en los resultados obtenidos al aplicar diversas técnicas de clusterización, se evidenció que, aunque la calidad de los clústeres no contaba con el mejor respaldo estadístico, el uso del tablero de Power BI, que contenía información sobre la población de clientes alertados en cada clúster y cómo estaban distribuidos, permitió una exploración funcional. Esta exploración permitió concentrar esfuerzos en grupos específicos donde se identificaron diferencias y características distintivas, como en los clústeres 23 y 39. Esta revisión funcional tomó aproximadamente 30 minutos y significativamente mejoró el proceso. Además, el tiempo promedio de gestión de una alerta se redujo a un mínimo de 4 horas. Con el agrupamiento resultante de los clústeres mencionados, se gestionaron 536 alertas en aproximadamente 8 horas.

9. CONCLUSIONES

- Para la selección de variables fue fundamental conocer a fondo cómo se gestiona una alerta en el área de investigación con el fin de abordar todas las características que debe tener este proceso, debido a la particularidad de como se gestiona una señal de alerta LAFT el resultado de nuestro dataset de entrenamiento presento una alta complejidad debido a la alta dimensionalidad y a la cantidad de variables que presentaban desbalanceo de clases.
- El mejor modelo de todos los modelos realizados con diferentes técnicas de clustering y con diferentes estructuras de datos (transformaciones), fue KMEANS con 5 componentes de los datos normalizados, debido a que en cuenta a las métricas de evaluación presento un mejor comportamiento y además el número de clústeres estaba más acorde a la definición funcional ya que reducía el riesgo de reprocesos y de pasar por alto particularidades en los clústeres formados que por la volumetría de alertas o clientes alertados en un solo grupo se puedan ignorar.
- El rendimiento de los modelos, las agrupaciones poco homogéneas resultantes de las diversas técnicas está muy ligado al tipo de problema ya que se presenta un enfoque LAFT el cual presenta características muy cambiantes en el tiempo además de la complejidad de la data.
- Siempre se debe contar con el apoyo funcional para la toma de decisiones en cuanto a la implementación de grupos formados garantizando una investigación suficientemente robusta y con una alta optimización en los tiempos presentando una mejora significativa en la oportunidad de gestión de señales de alerta transaccionales.

.

10. REFERENCIAS BIBLIOGRÁFICAS

[1] Infolaft, "infolaft.com," INFOLAFT. [Online]. Available: https://www.infolaft.com/lavado-de-activos-aumenta-el-numero-de-ros-en-colombia. [Accessed: 15-May-2023].

[2] Compliance, "Compliance sistema de información," Compliance sistema de información. [Online]. Available: https://www.compliance.com.co/que-es-un-ros/. [Accessed: 15-May-2023].

[3] Compliance sistema de información, "compliance.com.co," Compliance sistema de información. [Online]. Available: https://www.compliance.com.co/que-es-el-riesgo-la-ft/. [Accessed: 15-May-2023].

[4] GAFILAFT, "gafilat.org," 2016. [Online]. Available: https://www.gafilat.org/index.php/es/biblioteca-virtual/gafilat/documentos-de-interes-17/tipologias-17/353-recopilacion-tipologias-2010-2016/file. [Accessed: 20-May-2023].

[5] UIAF Unidad de Información y Análisis Financiero, "Módulo general - Curso: Lo que debe saber sobre lavado de activos y la financiación del terrorismo," Unidad de Información y Análisis Financiero UIAF, Bogota.

[6] Google for Developers, "k-Means Advantages and Disadvantages," Google for Developers. [Online]. Available: https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages. [Accessed: 26-Feb-2024].

[7] Aprende IA, "DBSCAN Teoría," Aprende IA. [Online]. Available: https://aprendeia.com/dbscan-teoria/. [Accessed: 26-Feb-2024].

[8] ArcGIS Pro, "Cómo funciona el clustering basado en densidad," ArcGIS Pro. [Online]. Available: https://pro.arcgis.com/es/pro-app/latest/tool-reference/spatial-statistics/how-density-based-clustering-works.htm. [Accessed: 26-Feb-2024].

[9] A. Fernández, "DBSCAN en Python: aprende cómo funciona," Ander Fernández. [Online]. Available: https://anderfernandez.com/blog/dbscan-python/. [Accessed: 26-Feb-2024].

[10] ImpulsateK, "DBSCAN: un algoritmo para detectar anomalías," ImpulsateK - Artificial intelligence, tools, insights and wisdom gleaned from the knowledge of others. [Online]. Available: https://impulsatek.com/dbscan-un-algoritmo-para-detectar-anomalias/. [Accessed: 26-Feb-2024].

[11] D. R. Yehoshua, "Spectral Clustering," Medium. [Online]. Available: https://medium.com/@roiyeho/spectral-clustering-50aee862d300. [Accessed: 26-Feb-2024].

[12] N. Doshi, "Spectral clustering," Medium. [Online]. Available: https://towardsdatascience.com/spectral-clustering-82d3cff3d3b7. [Accessed: 26-Feb-2024].

[13] Analytics Vidhya, "Spectral Clustering: A Comprehensive Guide for Beginners," Analytics Vidhya. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/what-why-and-how-of-spectral-clustering/. [Accessed: 26-Feb-2024].

[14] GeeksforGeeks, "Spectral Clustering in Machine Learning," GeeksforGeeks. [Online]. Available:

https://www.geeksforgeeks.org/ml-spectral-clustering/. [Accessed: 26-Feb-2024].

[15] E. G. PhD, "Affinity Propagation: Unveiling Clustering Patterns through Message Passing," Medium. [Online]. Available: https://medium.com/@evertongomede/affinity-propagation-unveiling-clustering-patterns-through-message-passing-eff3095eae72. [Accessed: 26-Feb-2024].

[16] GeeksforGeeks, "Affinity Propagation," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/affinity-propagation/. [Accessed: 26-Feb-2024].

[17] Machine Learning Explained, "Mean Shift," Machine Learning Explained. [Online]. Available: https://ml-explained.com/blog/mean-shift-explained. [Accessed: 26-Feb-2024].

[18] S. Dhumne, "Mean-Shift Clustering: A Powerful Technique for Data Analysis with Python," Medium. [Online]. Available: https://medium.com/@shruti.dhumne/mean-shift-clustering-a-powerful-technique-for-data-analysis-with-python-f0c26bfb808a. [Accessed: 26-Feb-2024].

[19] R. V. Escudero, "Clustering / Mean Shift. Métodos de Segmentación," Medium. [Online]. Available: https://medium.com/@a01793132/clustering-mean-shift-métodos-de-segmentación-f5523da4f539. [Accessed: 26-Feb-2024].

[20] Datanovia, "Agglomerative Hierarchical Clustering - Datanovia." [Online]. Available: https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/#algorithm.

[21] C. R. Patlolla, "Understanding the concept of Hierarchical clustering Technique," Medium, 10-Dec-2018. [Online]. Available: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec.

[22] GeeksforGeeks, "ML | OPTICS Clustering Explanation," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/ml-optics-clustering-explanation/. [Accessed: 26-Feb-2024].

[23] "One moment, please...," Datarundown. [Online]. Available: https://datarundown.com/optics-clustering/.

[24] R. R. II, "OPTICS CLUSTERING (Intro)," Medium, 25-Jun-2021. [Online]. Available: https://bobrupakroy.medium.com/optics-clustering-intro-76dcdaf94bde.

[25] JanbaskTraining, "Understanding OPTICS clustering: Identify the Clustering Structure." [Online]. Available: https://www.janbasktraining.com/tutorials/optics-clustering-algorithm/.

[26] P. K. K., "BIRCH Clustering Method: A Comprehensive Guide for Data Scientists in DNASeq and Variants Analysis," LinkedIn, 8-Aug-2023. [Online]. Available: https://www.linkedin.com/pulse/birch-clustering-method-comprehensive-guide-data-kandavel-phd.

[27] N. J. Benzer, "BALANCED ITERATIVE REDUCING AND CLUSTERING USING HEIRARCHIES (BIRCH)," Medium. [Online]. Available: https://medium.com/@noel.cs21/balanced-iterative-reducing-and-clustering-using-heirachies-birch-5680adffaa58. [Accessed: 26-Feb-2024].

[28] Javatpoint, "BIRCH in Data Mining." [Online]. Available: https://www.javatpoint.com/birch-in-data-mining.

[29] A. Z. Galiana, "repositori.udl.cat," Jul. 2021. [Online]. Available: https://repositori.udl.cat/items/2c79209