Sistema de recomendación personalizado basado en clustering aplicado a una empresa de fragancias



Aura María Atuesta Jiménez

Pontificia Universidad Javeriana
Tutor: Vladimir Moreno Gutierrez
Facultad de ciencias
Carrera de ciencias de datos
Trabajo de grado
2026-10

Índice general

1. Introducción **5**

1.1. Objetivo general . . . 6

1.2. Objetivos específicos . . . 6

2. Marco teórico **7**

2.1. Aprendizaje automático . . . 7

2.2. Aprendizaje no supervisado . . . 7

2.3. Modelo RFM . . . 8

2.4. Clustering . . . 8

2.4.1. Revisión de literatura del modelo RFM más clustering . . . 9

K-means . . . 9

Agrupamiento jerárquico . . . 11

DBSCAN . . . 13

BIRCH . . . 14

GMM . . . 15

Fuzzy C-means . . . 16

2.5. Sistemas de recomendación . . . 17

2.5.1. Principales enfoques . . . 17

2.5.2. Matriz de utilidad . . . 18

2.5.3. Revisión de literatura PRS para datos implícitos . . . 18

2.5.4. Medida de similitud . . . 18

2.5.5. Filtrado basado en contenido . . . 20

2.5.6. Enfoques basados en memoria . . . 21

Filtrado colaborativo basado en usuarios . . . 21

Filtrado colaborativo basado en artículos . . . 22

2.5.7. Enfoques basados en modelos . . . 22

2.5.8. Filtrado híbrido . . . 23

2.5.9. Retos de los RS . . . 23

2.5.10. Consideraciones para los RS . . . 25

2.5.11. Propuesta de implementación de técnicas de clusterización para la recomendación personalizada de productos para una plataforma de e-commerce en Honduras . . . 25

Recolección de datos . . . 25
Preprocesamiento de datos . . . 25
Implementación del modelo . . . 25
Resultados . . . 26
Limitaciones . . . 26
Conclusiones . . . 26
2.6. Relación entre clustering y RS . . . 26

3. Metodología **27**
3.1. Enfoque . . . 27
3.2. Alcance . . . 28
3.3. Contexto de la investigación . . . 28
3.4. Casos, universo y muestra . . . 28
3.5. Variables del estudio . . . 29
3.5.1. Historial de ventas . . . 29
Tipos de variables . . . 29
3.5.2. Fichas técnicas . . . 30
Tipos de variables . . . 31
3.5.3. Selección y limpieza . . . 31
Historial de ventas . . . 31
Fichas técnicas . . . 32
Modelo RFM . . . 33
3.6. Diseño . . . 34
3.7. Herramientas . . . 36
3.8. Procedimiento . . . 37
3.8.1. Clustering . . . 37
Hiperparámetros de los algoritmos . . . 38
Métricas de evaluación . . . 39
3.8.2. PRS . . . 40
Entrenamiento por clúster . . . 41
Filtrado basado en contenido . . . 42
UBCF . . . 42
IBCF . . . 42
HRS . . . 43
Recomendación popular . . . 43
3.8.3. Puntaje de las recomendaciones . . . 43
Validación cruzada . . . 45
3.8.4. Métricas de evaluación . . . 45
3.9. Consideraciones éticas y técnicas . . . 46

4. Resultados **47**
4.1. Variables del estudio . . . 47
4.1.1. Valores nulos . . . 47
4.1.2. Valores atípicos . . . 47
4.1.3. Distribución . . . 48
4.1.4. Análisis descriptivo de los clientes y artículos . . . 50
4.1.5. Análisis geográfico . . . 51
4.2. Modelo RFM . . . 51
4.3. Modelado . . . 53
4.3.1. Clustering . . . 53
Transformación de las variables RFM más otras . . . 53
PCA . . . 54
Clustering seleccionado . . . 55
Análisis de clustering . . . 56
4.3.2. PRS . . . 57

5. Conclusiones **60**

Bibliografía **62**

Capítulo 1

Introducción

La empresa de fragancias cuenta con dos tipos de líneas de productos en su catálogo. En primer lugar, se encuentran las fragancias industriales, las cuales se venden a empresas dedicadas a crear productos para la limpieza del hogar, aromatización ambiental, polímeros y/o cuidado personal. Y en segundo lugar, la perfumería fina para uso personal.
La Asociación Nacional de Empresarios de Colombia *(ANDI*, 2025) presentó resultados sobre el gasto de los hogares colombianos en el año 2024. La categoría de productos de belleza y de cuidado personal fue de aproximadamente US$2.884 millones, mostrando un aumento del 4,6 % frente al 2023. Por otro lado, los productos de aseo y cuidado del hogar tuvo un incremento del 7,4 % frente al año anterior, alcanzando los $20,62 billones de pesos. Estos datos demuestran que las personas están más interesadas en invertir en su cuidado personal y del hogar, incentivando la implementación de estrategias que fortalezcan su presencia en el mercado.
La empresa presenta un gran desafío debido a la variedad de fragancias, las opciones de perfiles olfativos, usos, intensidades, etc. Al no contar con un método que le permita ofrecer a un cliente nuevo o recurrente una experiencia de compra personalizada adaptada a sus gustos, pierde la oportunidad de tener ventajas competitivas y la recompra del cliente. A menos que el cliente tenga claridad sobre su orden de compra o se contacte con un vendedor y tenga un panorama general de los productos de la empresa.
Empresas globales como L'Oreal (L'Oréal, 2025), Sephora (Sephora newsroom, 2026) y Perfume's Club (News fragancias, 2022) cuentan con aplicaciones web donde cualquier persona puede ingresar y escoger sus marcas de preferencia en perfumería, las ocasiones de uso (día, noche), los perfumes que usa y los perfiles olfativos (cítrico, floral, dulce, etc.). Después de escoger, el cliente obtiene tres recomendaciones personalizadas de perfumes que no ha comprado. Permitiendo que descubra nuevas marcas y perfiles olfativos, sin salirse de sus gustos.
Se ha demostrado que los sistemas de recomendación personalizados impactan de manera positiva en la fidelización y la experiencia de los clientes, al igual que en el incremento de las ventas (De Biasio *et al.*, 2023), al realizar recomendaciones significativas para los clientes y rentables para las empresas. También ayudan a reducir la sobrecarga cognitiva, ya que facilitan la toma de decisiones. Iyengar y Lepper (2000) confirman que al reducir la cantidad de opciones de un catálogo, las personas se sienten más seguras al realizar compras. Esta herramienta se

ha implementado en sectores como bibliotecas, comercio electrónico y streaming, demostrando mejoras en la experiencia del cliente y en el dinamismo de los sectores (Tian *et al.*, 2019).

Se evidencia que con el avance tecnológico se han diseñado sistemas de recomendación personalizados que ayudan a que los clientes se sientan atendidos, conozcan nuevos productos, tengan una mejor experiencia de compra y al mismo tiempo se incrementen las ventas de las empresas. En este trabajo se presenta la oportunidad de implementar un sistema de recomendación personalizado a través del análisis del comportamiento de compra de los clientes por medio de sus interacciones con las fragancias y sus características, buscando realizar recomendaciones que fortalezcan la relación entre la empresa y sus clientes.
Con base en lo anterior, se presentan los siguientes objetivos que guiarán la elaboración de este trabajo.

1.1. Objetivo general

Desarrollar un sistema de recomendación personalizado para los clientes de una empresa del sector de fragancias, basado en técnicas de clustering aplicadas a la segmentación de las ventas históricas y las fichas técnicas de los productos, con el fin de personalizar la compra de los clientes, incrementar la fidelización y las ventas.

1.2. Objetivos específicos

1. Analizar y preprocesar los datos históricos de las ventas, aplicando técnicas estadísticas para identificar la estructura, distribución y datos faltantes de las variables.

2. Construir el modelo RFM (Recencia, Frecuencia, Valor Monetario) y aplicar métodos estadísticos que permitan transformar e identificar las variables críticas para segmentar el comportamiento de compra de los clientes.

3. Evaluar y comparar el rendimiento de los algoritmos de clustering mediante métricas de evaluación.

4. Estudiar los sistemas de recomendación y generar para cada clúster recomendaciones personalizadas teniendo en cuenta las fichas técnicas de los productos.

5. Evaluar el rendimiento de los sistemas de recomendación mediante métricas de evaluación.

A continuación, el capítulo 2 corresponde al marco teórico, donde se desarrollan los conceptos fundamentales. El capítulo 3 contiene la metodología propuesta. El capítulo 4 presenta los resultados obtenidos. El capítulo 5 contiene las conclusiones. Finalmente, se incluye el apartado bibliográfico donde se encuentran las fuentes que respaldan la construcción del trabajo.

Capítulo 2

Marco teórico

El presente trabajo de grado se basa en dos modelos: clustering y los sistemas de recomendación. Primero se abordan conceptos de aprendizaje automático, para darle paso al aprendizaje no supervisado, luego se presenta el modelo RFM acompañado del clustering con sus métodos, después los sistemas de recomendación con los tipos de filtrados y finalmente un estudio que integra ambas técnicas.

2.1. Aprendizaje automático

El aprendizaje automático o aprendizaje de máquina *(Machine Learning, ML)* es una rama de la inteligencia artificial *(Artificial Intelligence, AI)*. ML aprende de la experiencia y mejora su rendimiento a través de algoritmos y modelos estadísticos para extraer e identificar patrones en los datos. Su objetivo es realizar predicciones o tomar decisiones inteligentes con la información que tiene (Bishop, 2006).
Los tipos de ML tradicionales son, aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo (Bishop, 2006). En este trabajo se abordará solamente el aprendizaje no supervisado.

2.2. Aprendizaje no supervisado

Bishop (2006) expone que el algoritmo trabaja y aprende a partir de datos sin etiquetar. Su objetivo es identificar patrones o relaciones en los datos. Las categorías que usa son el agrupamiento y la reducción de dimensionalidad, esto permite que las máquinas aprendan a partir de patrones ocultos.
Su aplicación es común para encontrar datos atípicos *(outliers)*, simplificar los datos sin perder mucha información, agrupamiento o reglas de asociación.
Sin embargo, al trabajar con datos sin etiquetar, los resultados son impredecibles y se requiere un mayor análisis para comprender los resultados del algoritmo y así garantizar la calidad del agrupamiento o relación encontrada.

2.3. Modelo RFM

El modelo RFM (Recencia, Frecuencia y Valor monetario) permite a las empresas categorizar a los clientes según sus comportamientos de compra a través de los datos transaccionales, siendo un modelo clave para implementar estrategias de marketing.

Para Huang (2025) el análisis RFM se realiza para clasificar a los clientes en grupos, donde la recencia (R) corresponde al tiempo transcurrido entre la compra más reciente de un cliente y la fecha máxima registrada en los datos. Es importante interpretar los resultados de esta variable, un valor alto en R significa que han pasado ese número de días sin realizar una transacción, indica poca actividad del cliente. La variable de frecuencia (F) corresponde al número de compras realizadas durante el tiempo que se esté analizando, un valor alto en F significa que realiza varias transacciones (compras), indicando lealtad y gusto hacia el servio, producto, etc. Finalmente, valor monetario (M) corresponde al monto total de dinero gastado por el cliente en ese tiempo analizado.

El objetivo del modelo es ayudar a las empresas a identificar a sus clientes más valiosos e implementar estrategias para mejorar la fidelidad y permanencia del cliente (Gomes y Meisen, 2023). También permite identificar comportamientos interesantes, por ejemplo, un cliente tiene un valor en M alto pero en F bajo, esto indica que el cliente compra pocas veces pero cuando lo hace es en cantidades muy grandes o adquiere los productos/servicios más costosos.

Como afirma Huang (2025) varios estudios han ampliado el modelo RFM a otras aplicaciones, como la segmentación promocional, la clasificación de clientes bancarios y la recomendación de productos. Demostrando su utilidad para tener un mejor conocimiento de los clientes y su retención.

2.4. Clustering

Análisis de conglomerados *(clustering)* es una técnica estadística del aprendizaje no supervisado, se basa en agrupar observaciones o variables con características similares, busca que cada grupo *(clúster)* internamente sea semejante y a su vez diferente frente a los demás clústeres que se formen. El clustering ayuda a identificar comportamientos o patrones, es muy útil cuando se tiene un conjunto de datos grande, ya que permite encontrar respuestas a temas que no son sencillos a simple vista. Un ejemplo, al analizar datos de tráfico se logra observar en qué zonas se genera mayor tráfico o en qué horario suele ser más difícil el transporte. Permitiendo crear visualizaciones sencillas y ver las propiedades de los datos como la densidad, las relaciones entre clústeres, encontrar valores atípicos y saber por qué se pueden producir, pueden ser un error de digitación o un dato realmente muy diferente a los demás. Permitiendo reducir la complejidad de los datos al definir grupos con propiedades en común y entender el comportamiento, para un posterior uso de técnicas de reducción de dimensionalidad (IBM, 2025).

La partición de los datos se realiza utilizando la matriz de relación entre las variables o la matriz de distancias o similitudes, siempre y cuando cumpla con los siguientes supuestos (Peña, 2002):

1. Cada elemento debe pertenecer a solo un grupo.

2. Cada elemento debe ser parte de un grupo.

3. Cada clúster debe ser similar internamente.

Existen diferentes algoritmos que permiten realizar un enfoque específico para crear los clústeres. Como primer paso para cada algoritmo que se implemente se necesita saber el contexto a trabajar, qué pregunta se quiere responder, composición de los datos, tamaño de entrada, dimensionalidad, rigidez de las categorías, si se sabe el número de conglomerados que se desean formar y evaluación del agrupamiento. Chaudhry *et al.* (2023) realizó un diagrama de árbol de los métodos más utilizados para agrupar datos.

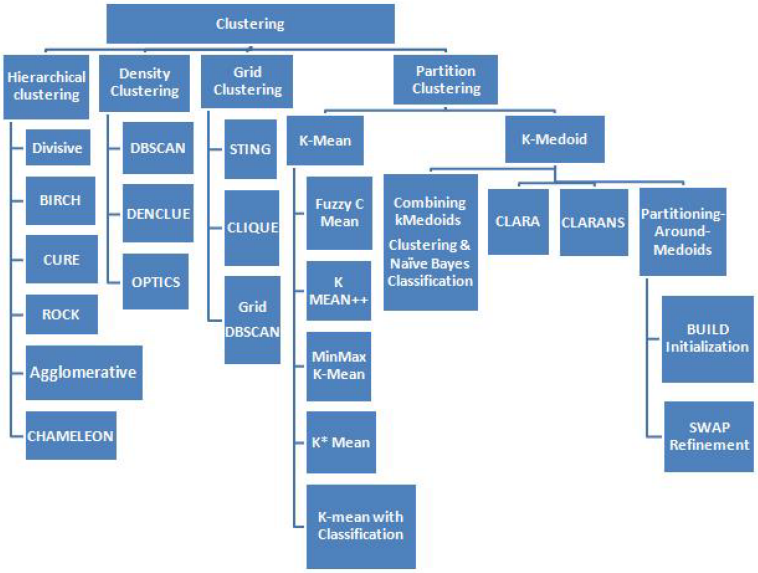

Figura 2.1: Jerarquía de los enfoques de agrupamiento (Chaudhry *et al.*, 2023)

2.4.1. Revisión de literatura del modelo RFM más clustering

Varios estudios han implementado el modelo RFM para analizar el comportamiento de compra de los clientes y posteriormente han segmentado estas variables con el fin de identificar los perfiles de los clientes y personalizar campañas de marketing. Cada estudio se realiza para un sector en el comercio electrónico, con un objetivo y un algoritmo de clustering donde se obtuvieron los mejores resultados a través de las métricas de evaluación y la interpretación de cada segmento. A continuación, se construye el cuadro 2.1 con los estudios relacionados que servirán como guía para tomar decisiones en el conjunto de datos a trabajar.

K-means

En la revisión de los métodos de segmentación de clientes realizada por Gomes y Meisen (2023), se concluyó que el algoritmo K-means es el más utilizado por su simplicidad y facilidad de implementación.
El libro de Raschka *et al.* (2022) permite entender que el objetivo del algoritmo es agrupar los

Cuadro 2.1: Estudios con el modelo RFM más segmentación de los clientes con clustering

Estudio	Yan et al., 2022	Anitha y Patil, 2022	Chen et al., 2012	Rungruang et al., 2024	Gomes y Meisen, 2023	Dedi et al., 2019	Chavhan et al., 2025	Christy et al., 2021
Dataset usado	199,002 registros (agosto 2021-julio 2022).	541,909 registros de datos transacciones	3,726 registros de transacciones del Reino Unido	5,878 clientes del UCI Machine Learning Repository	105 publicaciones científicas sobre segmentación de clientes en e-commerce (2000-2022)	174 clientes de productos de madera (2015-2018)	106,836 registros transaccionales de India	Datos transaccionales de UCI Machine Learning Repository
Objetivo	Caracterizar el comportamiento de carga para crear planes de configuración y campañas de marketing más efectivos	Identificar clientes potenciales y segmentar la base para mejorar las ventas y estrategias de marketing	Mejorar la comprensión de los clientes para facilitar estrategias de marketing y maximizar la rentabilidad.	Desarrollar un algoritmo de clustering FCA para segmentar a los clientes con conocimiento explícito e implícito	Investigación sobre los procesos de segmentación de clientes y su aplicación en marketing, enfocado en métodos de clustering y su efectividad en el análisis del comportamiento del cliente	Identificar tipos de clientes y medir su valor para determinar quiénes generan mayores beneficios y estrategias de marketing	Segmentar clientes usando RFM y una variable más (delivery ratio)	Segmentar en valores RFM y comparar el rendimiento de algoritmos de clustering duros (K-means) frente a difusos (Fuzzy C-Means).
Transformación de variables	NA	logarítmica	NA	Conversión de scores RFM a contexto formal binario (15 atributos)	Paso más importante	NA	Agrega una variable más llamada delivery ratio	NA
Normalización/ Escalamiento	Z-Score	NA	Min–Max	Método de quintiles para scores RFM (escala 1-5)	Paso más importante	Se realizó pero no menciona que técnica	Z-score	Min–Max
Algoritmo de clustering	BIRCH	K-means	K-means	FCA (Formal Concept Analysis) jerárquico vs K-means y clustering jerárquico tradicional	K-means: Principalmente utilizado para segmentación. Enfoques Híbridos: Combinaciones de diferentes métodos de clustering. Fuzzy C-Means: Para asignación de puntos a múltiples clústeres. Modelos de clase latente: Para clasificaciones más complejas.	K-means	GMM, K-means, DBSCAN, Agglomerative, Fuzzy C-Means	K-means, Fuzzy C-Means y Repetitive Median K-means (RM K-means)
Métricas / resultados	CH	Coeficiente de silueta = 0,362052482	La elección del número de clústeres se hizo a partir del análisis comparativo de los resultados (3, 4 y 5 clústeres)	Coeficiente de silueta y DB para evaluar el rendimiento **FCA** mostró ser el mejor ya que necesita menos preprocesamiento, no es sensible a valores atípicos y tiene gran capacidad de generar clústeres jerárquicos con conocimiento explícito e implícito	ANOVA Coeficiente de silueta Índices : DB, CH y Xie-Beni (XB)	Coeficiente de silueta = 0,7210 SSE (Sum of Square Error) = 7,3685	**GMM** obtuvo mejores resultados: Coeficiente de silueta = 0,7212 CH=22478.21 DB=0.654	**RM K-means** obtuvo los mejores resultados: Coeficiente de silueta = 0.49 Mostró una menor cantidad de iteraciones y un tiempo de ejecución más eficiente

datos en un número de clústeres *(identificado como K)* previamente definido, ya que su valor determina el enfoque y el detalle que se quiere tener dentro de cada clúster. Por lo tanto, el proceso inicia seleccionando K puntos al azar para que sean los centroides iniciales en el conjunto de datos. Se calcula la distancia desde los centroides definidos hasta los demás puntos y así se crean clústeres que indiquen la pertenencia de cada punto. Se desea obtener la mínima distancia entre los puntos y los clústeres asignados, donde los puntos más cercanos a un centroide hacen parte de un clúster de la misma categoría (Peña, 2002).
Ramírez (2024) destaca que el algoritmo funciona de forma iterativa, después de clasificar todos los puntos a un clúster, se reubican los centroides en el centro de cada grupo y se recalcula la distancia de cada punto a los centroides, hasta que los centroides ya no cambien o hasta un número de iteraciones definido por el programador. En caso de que entre un nuevo punto al conjunto de datos se recalculan las coordenadas del centroide del grupo al que se asigna.
Valor óptimo de K: El método tradicional que permite identificar el número óptimo de K es el método del codo, también conocido como inercia o WCSS (Within-Cluster Sum of Squares).

- **Método del codo**: Se grafican los valores de K y la suma del error cuadrático *(Sum of Squared Errors, SSE)* para cada valor de K (Rodríguez, 2023). SSE es la suma de la distancia de cada punto al centroide del clúster. Por lo tanto, cuando SSE disminuye más lento es el punto óptimo de K, se conoce como el codo. Por ejemplo, en la Figura 2.2 el número óptimo y donde se forma el codo es en $K = 3$.

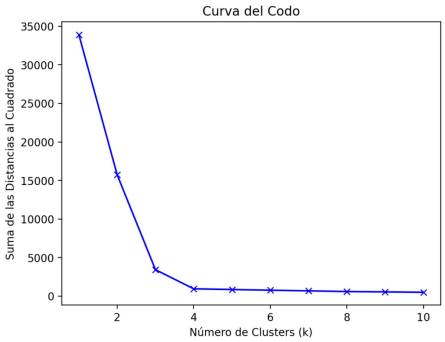

Figura 2.2: Método del codo (Rodríguez, 2023)

Desafíos: El algoritmo K-means no funciona bien cuando los datos tienen altas dimensiones o los clústeres tienen tamaños o densidades muy diferentes (IBM, 2025). También se sobreajusta con los valores atípicos, ya que se dificulta el cálculo de las distancias, sesgando el algoritmo con valores muy grandes.

Agrupamiento jerárquico

Construye subconjuntos de clústeres organizados jerárquicamente como un árbol, llamado dendrograma (Peña, 2002). Se compone del eje horizontal donde van las etiquetas de las observaciones que se están agrupando y el eje vertical con la distancia entre las observaciones o clústeres (Noble, 2025). Como se observa en la Figura 2.3, cada nodo representa los puntos de

unión de dos clústeres, la altura muestra el nivel de similitud y los colores muestran los clústeres formados.

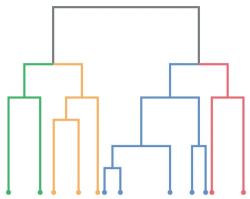

Figura 2.3: Dendrograma simple (Shutterstock, s.f.)

Julio *et al.* (2025) explican que existen dos tipos de enfoques para el agrupamiento jerárquico: el aglomerativo y el de división.

- **Aglomerativo:** Es un algoritmo conocido por la sigla AGNES *(Aglomerative nesting)* introducido por Kaufman y Rousseeuw (1990), supone que cada individuo es un clúster y se van formando en grupos según su proximidad. Se observa en la Figura 2.4 el paso a paso que sigue el algoritmo.

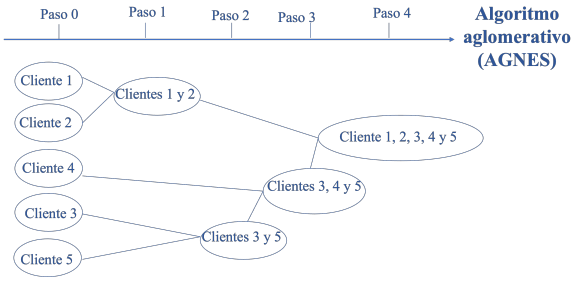

Figura 2.4: Algoritmo AGNES (Julio *et al.*, 2025)

  El primer paso es construir una matriz cuadrada y simétrica de distancias, cada fila y columna contiene la distancia calculada entre cada uno de los puntos de las observaciones (Peña, 2002). La medida de distancia que puede ser, euclidiana, Mahalanobis, Manhattan, Minkowski, entre otras. Después de calcular las distancias se escoge un método de aglomeración, donde cada individuo de manera iterativa va a pertenecer a un clúster, existe enlace único, completo, promedio, mediano, centroide, Ward, McQuitty, entre otros.
  En la práctica es difícil saber cuál de estos métodos generará grupos óptimos, recomiendan iterar con diferentes combinaciones entre los métodos y las distancias para identificar cuál es el mejor número de conglomerados.

- **División:** Es un algoritmo conocido por la sigla DIANA *(Divise analysis)* introducido por Kaufman y Rousseeuw (1990). A diferencia del método aglomerativo, comienza con un único clúster, este contiene todas las observaciones (puntos), su objetivo es generar divisiones hasta que cada observación forme parte de un grupo independiente. Se observa en la Figura 2.10 el paso a paso que sigue el algoritmo.

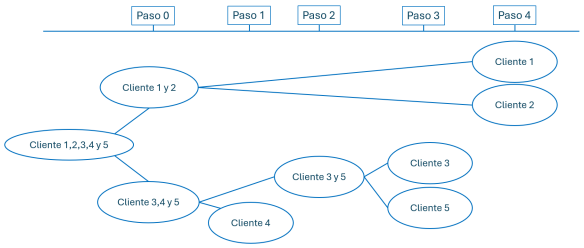

Figura 2.5: Algoritmo DIANA

Se basa en seleccionar el clúster con la distancia más grande (mayor diámetro) entre dos puntos dentro de él. Este algoritmo solo trabaja con distancias que pueden ser las mismas que se usan para el aglomerativo y a diferencia del aglomerativo no necesita métodos de aglomeración.

Después de seleccionar el clúster con mayor diámetro, se busca la observación que tenga la mayor distancia promedio respecto a las demás y crea un nuevo grupo, si las demás observaciones están más cerca al nuevo grupo creado se puede reasignar a ese o permanecer con el grupo original. Este proceso se repite hasta que cada individuo quede en un clúster independiente.

Desafíos: Noble (2025) comenta que cuando el clustering jerárquico tiene muchas observaciones el dendrograma tiene demasiadas ramas y se complica su visualización e interpretación. El aglomerativo también se ve afectado por el proceso computacional por la cantidad de distancias y métodos a calcular. Otro desafío es cortar el dendrograma y decidir el número de clústeres, sin embargo, se cuenta con métricas de evaluación.

DBSCAN

Es un algoritmo de agrupamiento basado en la densidad *(Density-Based Spatial Clustering of Applications with Noise)* introducido por Ester *et al.* (1996). El algoritmo es capaz de identificar puntos muy cercanos, teniendo en cuenta el espacio que haya entre ellos. Se especializa en manejar valores atípicos *(outliers)*, ya que no los integra a un clúster y los deja por fuera. Es bueno manejando irregularidades en los datos ya que se adapta a cualquier forma para crear los clústeres y no necesitan ser circulares o convexos (GeeksforGeeks, 2019).
Se requiere la definición de tres parámetros para su uso:

1. Épsilon (ϵ): Distancia máxima entre dos muestras para que cada una sea considerada como un clúster de la otra, es el radio alrededor de los puntos de datos (Bomba, 2023). Este valor es el más importante, un valor demasiado pequeño clasificaría como ruido y uno demasiado grande puede tomar a todos los puntos como un clúster y no distinguirlos (GeeksforGeeks, 2019).

2. MinPts: Número mínimo de puntos requeridos dentro del radio dado por ϵ para formar un clúster. Generalmente se tiene MinPts $\geq D + 1$, siendo D el número de dimensiones del conjunto de datos (GeeksforGeeks, 2019).

3. Métrica: Calculo de la distancia entre dos puntos, puede ser la euclidiana u otras métricas de proximidad (Bomba, 2023).

Raschka *et al.* (2022) explican que el algoritmo categoriza los datos en tres tipos de puntos:

1. Punto central: Tiene un número suficiente de vecinos dado por el usuario en MinPts y dentro de un radio ϵ.

2. Punto fronterizo: Está cerca del punto central pero alrededor de ese punto contiene menos puntos de MinPts, careciendo de vecinos para ser un punto central.

3. Punto de ruido: No pertenece a ningún clúster, está fuera del ϵ y muy alejado de los otros puntos.

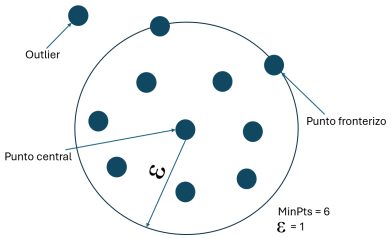

Figura 2.6: Ilustración del funcionamiento de DBSCAN

La construcción de los clústeres comienza seleccionando un punto al azar que no se haya visitado. Si ese punto es un punto central, se expande incluyendo todos los puntos que estén dentro de ϵ y que cumplan con el parámetro MinPts y este sería un clúster (Bomba, 2023).
En caso de que no sea un punto central se marca como ruido temporalmente, teniendo en cuenta que este proceso se repite hasta que todos los puntos queden marcados como punto central, fronterizo o ruido.
Desafíos: Escoger el número adecuado de ϵ y MinPts es muy importante para la expansión y el detalle de cada clúster. Tiene problemas para manejar conjuntos de datos de alta dimensión por los cálculos que debe realizar y de dimensión variable.

BIRCH

Este algoritmo fue introducido por Zhang *et al.* (1996) y se elige por su buen rendimiento en los resultados reportados en el estudio de Yan *et al.* (2022). Es un algoritmo de agrupamiento jerárquico incremental (*Balanced Iterative Reducing and Clustering Using Hierarchies, BIRCH*). Su estructura es un árbol, conocido como árbol de características de agrupamiento (CF) para todos los datos. Estos datos se comprimen con pérdida en un conjunto de nodos, los cuales tienen subgrupos y pueden tener nodos hijos. En la siguiente figura se muestra cómo se unen a través de un puntero que los enlaza en una lista.

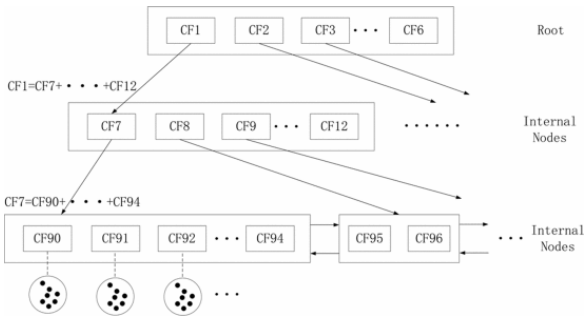

Figura 2.7: Árbol de características de BIRCH (Yan *et al.* (2022))

La característica de agrupamiento CF es un vector tridimensional que se define como: $CF = (N, LS, SS)$

- N: Número de muestras en el subgrupo.
- LS: Vector n-dimensional con la suma de todas las muestras. $\sum_{i=1}^{N} \vec{x}_i$
- SS: Suma de la norma Euclídea (L2) al cuadrado de cada muestra. $\sum_{i=1}^{N} \|\vec{x}_i\|^2$

Adicionalmente se necesitan:

- Centroides: Vector n-dimensional con la suma de todas las muestras sobre el numero total de muestras. $\frac{LS}{N}$
- Norma cuadrada de los centroides: Magnitud del vector centroide respecto al origen. $\left\| \frac{LS}{N} \right\|^2$

Es importante tener en cuenta que el algoritmo cuenta con dos parámetros: el *umbral*, este limita la distancia entre la muestra entrante y los subgrupos existentes y el *factor de ramificación*, limita el número de subgrupos en un nodo (scikit-learn, 2010).
En el proceso de agrupamiento, el algoritmo se compone de 4 fases (Han *et al.*, 2012). En la fase 1 se construye un árbol CF inicial, creando nodos. En la fase 2 se puede reagrupar para reducir aún más el tamaño del árbol, elimina ruido u outliers. En la fase 3 se trabaja con las hojas del árbol, se usan métodos de clustering tradicionales para definir la frontera de los grupos. Finalmente, la fase 4 utiliza los centroides encontrados en la fase 3 para reasignar cada punto del dataset original al clúster más cercano, esto sirve para corregir desviaciones que se pudieron pasar en la fase 1.

GMM

Este algoritmo fue introducido por Wolfe (1970) y se elige por su buen rendimiento en los resultados reportados en el estudio de Chavhan *et al.* (2025). De acuerdo con GeeksforGeeks (2018), es un modelo de mezcla gaussiana que supone que los datos se generan a partir de varias distribuciones gaussianas (normales), donde cada uno representa un clúster k. Por lo tanto, agrupa los puntos de datos que pertenecen a una misma distribución a través de probabilidades y permite crear clústeres con formas elípticas.

Cada clúster es una gaussiana $\mathcal{N}(\mu_k, \Sigma_k)$ que tiene su propia media μ_k y matriz de covarianza Σ_k.

Los modelos GMM se entrenan utilizando un proceso iterativo que estima los parámetros, llamado EM (Expectation-Maximization). Expectativa corresponde a calcular la probabilidad a priori π_k de que un punto pertenezca a un k, usando los parámetros actuales. Maximización tiene como objetivo actualizar μ_k, Σ_k y π_k con el paso de expectativa, el proceso continúa hasta que las iteraciones logren que la verosimilitud logarítmica del modelo se estabilice, siguiendo la siguiente fórmula:

$$L(\mu_k, \Sigma_k, \pi_k) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)$$

- N: Número total de puntos de datos.
- K: Número total de clústeres.
- π_k: Probabilidad a priori de pertenecer al clúster k

Fuzzy C-means

FCM se elige por su uso en los estudios de Gomes y Meisen (2023), Chavhan *et al.* (2025) y Christy *et al.* (2021). Santoyo *et al.* (2023) definen el algoritmo como difuso, ya que puede pertenecer a varios clústeres mediante un valor entre 0 y 1. El objetivo es minimizar la distancia euclidiana entre los centroides y los puntos de datos (como K-means), teniendo en cuenta el grado de difusividad m en el rango $[1, n]$ que determina la suavidad de la asignación de pertenencia de cada punto a los clústeres. Un valor alto en m (> 1) suaviza y permite que los puntos tengan una pertenencia más equitativa entre diferentes clústeres.

Para minimizar la función, el algoritmo itera actualizando m y los centroides, FCM se define:

$$\mu_{ij} = \left[\sum_{t=1}^{c} \left(\frac{\|x_j - v_i\|_A}{\|x_j - v_t\|_A} \right)^{\frac{2}{m-1}} \right]^{-1}$$

- μ_{ij}: Probabilidad de pertenencia del dato j al clúster i.
- c: Número total de clústeres.
- x_j: j-ésimo dato.
- v_i: i-ésimo centroide sobre el que se calculará el grado de pertenencia.
- $||_A$: Norma que mide qué tan lejos está el dato x_j del centroide v_i.
- $\sum_{t=1}^{c} \mu_{tj} = 1$: Restricción de normalización. La suma de los grados de membresía de cualquier grupo es uno.

2.5. Sistemas de recomendación

Los sistemas de recomendación *(Recommender Systems, RS)* son algoritmos computacionales que se encargan de filtrar información relevante de acuerdo con el comportamiento en compras, las reseñas y/o calificaciones a productos o servicios *(artículos)* de los clientes *(usuarios)*. El objetivo es realizar recomendaciones de acuerdo con los gustos del usuario teniendo en cuenta que sea, un artículo que no ha consumido antes, de su interés y diverso, es decir, que recomiende artículos variados y no se quede siempre en una misma línea (Aggarwal, 2016).
En los últimos años, los RS han tenido mayor visibilidad con el avance en plataformas en línea como el streaming, redes sociales, comercio electrónico, entre otros. Mostrando interés por parte de las empresas en crear modelos que conozcan a los usuarios y atiendan sus necesidades orientadas al campo de cada una.
Por ejemplo, actualmente los RS se aplican en plataformas de streaming con recomendaciones como "Dado que te gusta esta película...", "Ya que viste...", "Similares a ...". También se aplica en comercio electrónico con recomendaciones como "Dado que tu última compra fue...", "Es tu mes de cumpleaños...", "Otros usuarios también compraron..."
Este tipo de recomendaciones se llaman sistemas de recomendación personalizados (*Personalized Recommender Systems, PRS*), se basan en el comportamiento individual de cada usuario y se crean recomendaciones teniendo en cuenta sus preferencias demostradas a través de compras, consumo o calificación de los artículos. Otro caso de los RS son las conocidas como "Top list", son recomendaciones generales o populares que recomiendan lo que esté en tendencia, como la cantidad de me gusta que tenga algún artículo o el más consumido en el momento.

2.5.1. Principales enfoques

Los RS tienen dos tipos de retroalimentación, este es un punto clave para recopilar, trabajar e interpretar los datos:

- Explícito: Se entiende claramente la preferencia del usuario, es directo y se realiza a través de puntuaciones, comentarios o likes que da el usuario a los artículos. Existen diferentes retos, (1) los usuarios retroalimenten (califiquen o comenten) los artículos, (2) los usuarios sean honestos con esas retroalimentaciones y (3) el dueño del perfil sea el que realice la retroalimentación y no otra persona (Milvus, s.f.).

- Implícito: Se basa en el comportamiento de interacción que realice el usuario con los artículos, es decir, historial de compras, clics, búsquedas o tiempo dentro de la página (Isinkaye *et al.*, 2015). Existen diferentes retos, (1) interpretar correctamente las señales percibidas y (2) la interacción la realice el dueño del perfil y no otra persona (Milvus, s.f.).

Estas retroalimentaciones se utilizan para construir una matriz usuario-artículo llamada matriz de utilidad.

2.5.2. Matriz de utilidad

La matriz de utilidad se organiza con el artículo en las columnas y el usuario en las filas (Roy y Dutta, 2022). Mostrando las calificaciones que un usuario le da a cada artículo, como se muestra en el Cuadro 2.1.

Usuario / Artículo	**Artículo 1**	**Artículo 2**	**Artículo 3**	**Artículo 4**
Usuario 1	5	3	4,5	-
Usuario 2	2	3	-	-
Usuario 3	-	-	4	3,5
Usuario 4	1	-	3,7	4

Cuadro 2.2: Matriz de utilidad

En caso de no contar con las calificaciones pone la cantidad de artículos comprados, se realiza una matriz binaria indicando 1 compra 0 no, etc.

2.5.3. Revisión de literatura PRS para datos implícitos

En los últimos años varios estudios han implementado PRS basados en clustering, con el fin de reducir el costo computacional y conocer mejor las preferencias de cada usuario a través de los segmentos y perfiles que se pueden crear. Además, enfrentan retos como trabajar con datos implícitos, usuarios con gustos variados y sectores del comercio electrónico y streaming. A continuación, se construye el cuadro 2.3 con los estudios que trabajaron con datos implícitos, como resolvieron la matriz de utilidad (usuario–artículo), que clustering usaron, que modelo de RS implementaron y que métricas fueron útiles para validar los resultados obtenidos.

2.5.4. Medida de similitud

Se conoce como medida de similitud o de distancia, se encarga de medir la proximidad que hay entre dos distribuciones o conjuntos de datos (Roy y Dutta, 2022). En los RS se usa para calcular las puntuaciones que reflejan la similitud entre usuarios o artículos, esto permite resolver problemas de reconocimiento de patrones, como la clasificación y la agrupación (Eberly College of Science, 2018).

Un ejemplo sencillo aplicado a los RS es buscar un grupo de usuarios (X) cuyas preferencias, gustos o disgustos sean similares a los del usuario (A). X se conoce como el vecindario de A. Por lo tanto, los nuevos artículos que le gusten a la mayoría de los usuarios en X se le recomiendan al usuario A (Roy y Dutta, 2022).

Dentro de las dos medidas de similitud más populares están la basada en la correlación de Pearson que se utiliza para medir el grado de relación lineal entre dos variables (Isinkaye *et al.*, 2015), se define como:

$$s(a,u) = \frac{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i=1}^{n}(r_{a,i} - \overline{r_a})^2} \cdot \sqrt{\sum_{i=1}^{n}(r_{u,i} - \overline{r_u})^2}}$$

Cuadro 2.3: Estudios de sistemas de recomendación

Estudio	Chen et al., 2024	Stylianou y Pantelidou, 2025	Wang et al., 2020	Zamanzadeh y Valipour, 2022	Hwangbo et al., 2018	Shinde y Kulkarni, 2012	Darraz et al., 2025
Tipo RS	Sistema de recomendación colaborativo	Sistema de recomendación de filtrado colaborativo basado en artículos con la segmentación de usuarios	Sistema de recomendación de filtrado colaborativo basado en artículos	Sistema de recomendación híbrido basado en grafos que combina filtrado colaborativo y atributos de usuario	Sistema de recomendación colaborativo basado en artículos en el sector de moda	Sistema de recomendación híbrido personalizado que utiliza un algoritmo de agrupamiento basado en centrado.	Sistemas de recomendación híbridos con análisis de sentimientos
Tipo de datos	Datos implícitos del comportamiento del usuario (interacciones, clics, vistas, compras)	Datos implícitos transaccionales de supermercados	Datos explícitos de calificaciones históricas	Datos explícitos de calificaciones, demográficos y características estructurales de grafos de similitud	Datos implícitos (clics), demográficos y datos transaccionales	Datos explícitos de calificaciones, demográficos y datos transaccionales	Datos explícitos de calificaciones, comentarios, información de negocios (restaurantes y hoteles) y perfiles
Matriz usuario-artículo	Matriz que se descompone en dos matrices mediante factorización matricial (MF)	Matriz binaria donde 1 indica que el usuario compró el artículo y 0 que no lo compró	Matriz con diferentes niveles de densidad dependiendo del dataset trabajado: MovieLens y Netflix	Matriz complementada con representaciones latentes extraídas de grafos	Matriz con valores ponderados mediante función de decaimiento temporal	Matriz de calificaciones normalizada (0-1), 0 indica artículo no calificado	Matriz de factorización profunda de matrices (DeepMF) para capturar interacción usuario–artículo, complementada con representaciones latentes de texto y atributos
Uso de clustering	No, usa muestreo negativo basado en asociaciones entre artículos	Sí, usa K-means con k=5 para segmentar usuarios según patrones de compra por departamento	No, usa K-Nearest Neighbors (KNN) para encontrar artículos similares	Sí, usa K-means para agrupar usuarios usando autoencoder	Sí, usa K-means usando los patrones de ventas mensuales	Sí, usa Centering-Bunching Based Clustering (CBBC) y encontró tres clústeres.	Sí, clustering de artículos basado en texto (Word2Vec), StandardScaler y reducción de dimensionalidad (PCA). Encontró con K-means 11 clústeres para restaurantes y con jerárquico 12 para hoteles
Modelo principal	Modelo Bayesian Personalized Ranking (BPR) optimizado con Association-Based Negative Sampling (ABNS) y se usa la similitud coseno entre artículos	Filtrado colaborativo basado en artículos y en los segmentos de usuarios	Modelo de filtrado colaborativo basado en artículos, combina similitud Ochiai para estructural y similitud coseno para calificaciones en una función híbrida	Autoencoder de múltiples capas para extracción de características, más K-means y predice basado en los promedios de los clústeres	Filtrado colaborativo basado en artículos con decaimiento temporal, similitud coseno y combinación de datos online y offline	Para el preprocesamiento de los datos y clustering lo realiza offline y el PRS online	Modelo híbrido que combina DeepMF, análisis de sentimientos con BERT, reducción de dimensionalidad con factorización de matrices no negativas (NMF) y un DecisionTreeRegressor para fusionar predicciones basadas en contenido y sentimientos
División de datos	Aleatoriamente se divide en 80 % de los datos para entrenamiento y 20 % para prueba	Validación cruzada de 5 pliegues para entrenamiento y prueba	Validación cruzada de 5 pliegues para MovieLens y conjunto de prueba predefinido para Netflix	Se realizó validación cruzada de 5 y 10 pliegues dependiendo del dataset trabajado	Evaluación mediante prueba A/B durante 3 semanas en entorno real	Matriz de entrenamiento y evaluación sobre usuario activo como conjunto de prueba	70 % de los datos para entrenamiento y 30 % para prueba en el análisis de sentimientos. DeepMF usó validación cruzada de 8 pliegues
Métricas / Resultados	Con Recall@20 NDCG@20 respectiva-mente,Movielens100K: 0.3373, 0.4013, LastFM: 0.3100, 0.2856, Movielens2K: 0.1436, 0.4691, Tools: 0.0938, 0.0437	Precision = 87 %, AUC =0.89, F1-score=0.81, curva ROC	Con MAE (Mean Absolute Error) MovieLens: 0.7045 y Netflix= 0.7028	MovieLens 100K: RMSE = 0.887, Precisión = 0.771, Recall = 0.799 y MovieLens 1M: RMSE = 0.833, Precisión = 0.792, Recall = 0.838	El grupo experimental generó 9.9 % clics y 12.3 % compras por recomendaciones vs 5.8 % clics y 8.9 % compras del grupo control.	Precision = 0.9, Recall = 0.89 y MAE = 0.8	Restaurantes: RMSE = 0.02 y MAE=0.054. Hoteles: RMSE = 0.01 y MAE = 0.07

$s(a, u)$ es la similitud entre dos usuarios a y u.
$r_{a,i}$: Calificación dada al artículo i por el usuario a.
$\overline{r_a}$: Calificación media dada por el usuario a.
$r_{u,i}$: Calificación dada al artículo i por el usuario u.
$\overline{r_u}$: Calificación media dada por el usuario u.
n: Número total de artículos, dentro de la matriz de utilidad.

Y la similitud del coseno que se enfoca en un modelo espacio vectorial basado en álgebra lineal en lugar de un enfoque estadístico como lo hace Pearson. Mide la similitud entre dos vectores n-dimensionales basándose en el ángulo que los une (Isinkaye *et al.*, 2015), se define como:

$$s(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{n} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i=1}^{n} r_{u,i}^2} \cdot \sqrt{\sum_{i=1}^{n} r_{v,i}^2}}$$

$\vec{u}, \vec{v}$: vectores de usuarios.
$r_{u,i}$: Calificación dada al artículo i por el usuario u.
$r_{v,i}$: Calificación dada al artículo i por el usuario v.
n: Número total de artículos, dentro de la matriz de utilidad. O solamente los artículos calificados por los usuarios u y v.

Se obtienen valores de *similitudes* $\in [0, 1]$, donde *1* indica similitud total (Eberly College of Science, 2018).
Isinkaye *et al.* (2015) y Roy y Dutta (2022) comentan que los RS se componen en tres tipos de filtrados. Basado en contenido, híbrido y colaborativo con dos enfoques: memoria y modelos.

2.5.5. Filtrado basado en contenido

La técnica de filtrado basado en contenido extrae las características de los artículos que ha adquirido el usuario (Roy y Dutta, 2022). Crea una matriz de los artículos y sus características, con el objetivo de identificar la similitud de cada artículo con otro. Por ejemplo, un usuario compra un perfume, las características son: olor, uso, marca, etc. Dentro del sistema se busca la mayor similitud con otros perfumes que también tengan esas características para realizar la recomendación del artículo, como se muestra en la Figura 2.8.

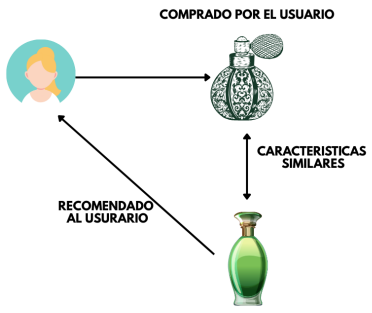

Figura 2.8: Filtrado basado en contenido

Este filtrado necesita tener conocimiento profundo sobre las características de cada artículo para hacer recomendaciones precisas (Roy y Dutta, 2022). Además, presenta dificultad para mostrar recomendaciones novedosas, ya que siempre se basa en las características puntuales de los artículos. Sin embargo, es eficiente ante el cambio de los gustos o compras de los usuarios, ya que se acopla a las nuevas características sin problema. Su mayor ventaja es que no necesita de las calificaciones de los clientes, solo las características de los artículos (Beregovskaya y Koroteev, 2021).

2.5.6. Enfoques basados en memoria

Se basa en realizar recomendaciones a través de la matriz de utilidad (Roy y Dutta, 2022), solo recomienda artículos a las personas que se encuentren en la matriz, por lo tanto, si llega un usuario nuevo hay que incluirlo en la matriz y recalcular la medida de similitud para realizar la recomendación. Siendo una técnica más compleja, empleada en el filtrado colaborativo basado en usuarios y el filtrado colaborativo basado en artículos.

Filtrado colaborativo basado en usuarios

La técnica de filtrado colaborativo basado en usuarios (*User-Based Collaborative Filtering, UBCF*), tiene como objetivo encontrar a otro(s) usuario(s) con calificaciones o compras similares. Después de encontrar usuarios con compras similares se recomiendan artículos al usuario objetivo que no ha consumido en comparación con los demás (Beregovskaya y Koroteev, 2021). Por ejemplo, se encontró que dos usuarios 1 y 2 (usuario 2 objetivo), ambos compran el perfume verde y rosado, sin embargo, el usuario 1 compra el perfume azul y usuario 2 no. Al encontrar similitud en sus compras se le recomienda el perfume azul al usuario 2, como se muestra en la Figura 2.9.

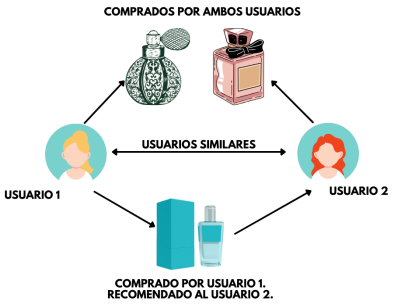

Figura 2.9: Filtrado colaborativo basado en usuarios

Esta técnica presenta problemas de privacidad al trabajar con datos de usuarios. Se deben tener medidas de seguridad para el filtrado de información y la manipulación inadecuada de los datos (Roy y Dutta, 2022). Sin embargo, su mayor ventaja es que no requiere del conocimiento de las características de los artículos y realiza recomendaciones novedosas a los usuarios al basarse en la similitud del perfil con otros (Beregovskaya y Koroteev, 2021).

Filtrado colaborativo basado en artículos

La técnica de filtrado colaborativo basado en artículos (*Item-Based Collaborative Filtering, IBCF*) tiene como objetivo encontrar otro (S) artículo (s) con calificaciones o comportamientos de compras similares. Los usuarios suelen comprar artículos al mismo tiempo aun cuando no comparten características, por ejemplo, compran dos perfumes cada que van al punto de venta el verde y amarillo. Para encontrar artículos con compras similares se tiene un artículo objetivo, las recomendaciones se realizan basándose en la similitud del artículo que muestra tener relación con el objetivo y no ha consumido (Roy y Dutta, 2022).

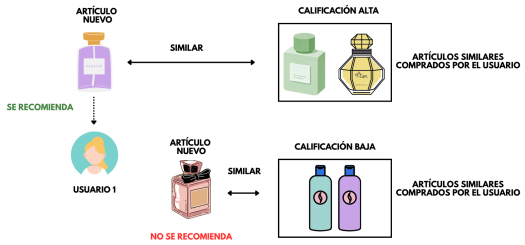

Figura 2.10: Filtrado colaborativo basado en artículos

Esta técnica no trabaja con la similitud entre usuarios, pero si busca patrones de compra entre ellos, no requiere del conocimiento de las características de los artículos. A diferencia del filtrado basado en contenido, su fuente de datos es el historial de compras a través de la matriz usuario–artículo, mientras que basado en contenido trabaja con los atributos de los artículos a través de las fichas técnicas (Isinkaye *et al.*, 2015).

2.5.7. Enfoques basados en modelos

Utiliza algoritmos de minería de datos y aprendizaje automático para desarrollar un modelo que prediga la calificación del usuario para un artículo sin calificación (Roy y Dutta, 2022). Extrae características de la matriz de utilidad para calcular las calificaciones, como la frecuencia de compra (Isinkaye *et al.*, 2015). Lo realiza en dos pasos:

1. Construir el modelo a través de las características de la matriz de utilidad.
2. Predecir las calificaciones tomando como entrada el modelo y el perfil del usuario que debe pertenecer a la matriz de utilidad.

Su ventaja es que si llega un nuevo usuario no se necesita agregar a la matriz (Roy y Dutta, 2022), se puede realizar la recomendación sin que pertenezca, permitiendo recomendar más rápido debido al modelo preentrenado. Resolviendo problemas de reducción de dimensionalidad, escasez y escalabilidad de los datos. El mayor reto es emplear el algoritmo de aprendizaje adecuado para la creación del modelo.

2.5.8. Filtrado híbrido

La técnica de filtrado híbrido (*Hybrid Recommender Systems*, *HRS*) combina diferentes técnicas de recomendación para superar los límites o problemas que presentan al usar solo una de ellas (Isinkaye *et al.*, 2015). La combinación puede tener diferentes filtros, implementaciones y combinaciones de los PRS, tal y como lo explican Isinkaye *et al.* (2015), el objetivo es mejorar las recomendaciones y métricas de evaluación al trabajar con HRS:

- **Hibridación ponderada:** Combina los resultados de diferentes filtros para generar una lista o predicción de recomendaciones, integra las puntuaciones de cada técnica usada mediante una fórmula lineal, ajustándose a medida que las predicciones son correctas.
- **Hibridación conmutada:** El sistema cambia a un filtrado específico según el problema que esté abordando. Es capaz de detectar problemas y actuar, por ejemplo, si se quiere incluir un nuevo usuario en un algoritmo que es basado en contenido, cambiaría a un sistema de recomendación colaborativo. Su desventaja es el aumento en el número de parámetros ya que dificulta la computación y los tiempos.
- **Hibridación en cascada:** Tiene un proceso iterativo para construir las preferencias entre diferentes artículos por parte de los usuarios. El primer filtro que aplica genera una lista de recomendaciones, después lo mejora comparando los resultados con el siguiente filtro que aplique. Como se trabaja con iteraciones tolera el ruido y genera recomendaciones más precisas.
- **Hibridación mixta:** Combina simultáneamente los resultados de las recomendaciones de diferentes filtros. Cada artículo tiene múltiples recomendaciones dadas por diferentes filtros, su objetivo es mostrar todas esas recomendaciones.
- **Aumento de características:** Los filtrados colaborativos y basados en contenido utilizan las calificaciones, información generada por el usuario e información adicional que se tenga de los artículos. Sin embargo, se puede acudir a otras fuentes de información donde se puede acceder a descripciones de los artículos. Se realiza mediante un clasificador de texto bayesiano ingenuo. Añadiendo un pequeño número de características al recomendador principal.
- **Metanivel:** Utiliza un modelo generado por algún filtro de recomendación como entrada para otro. Este método resuelve el problema de escasez, ya que al utilizar un modelo aprendido se tiene más información valiosa para el segundo modelo que se implemente.

2.5.9. Retos de los RS

La revisión de literatura realizada por Roy y Dutta (2022) y el libro de Aggarwal (2016), identifican problemas y soluciones que garantizan su minimización al desarrollarlos, tales como:

- **Inicio en frío:** Sucede cuando no hay datos suficientes sobre las interacciones de los usuarios, sobre todo cuando se une un nuevo usuario y no se tiene historial de sus compras o calificaciones, por lo tanto, los RS no logran encontrar otro usuario similar. Esta situación también ocurre cuando se agrega un nuevo artículo y ningún usuario ha interactuado con este. En general, el sistema lo oculta y no genera recomendaciones.
  Posible solución: Solicitar al usuario (1) llenar una encuesta con artículos de su pereferencia o (2) calificar algunos artículos. Otra opción será realizar recomendaciones populares.

- **Ataque de Shilling:** Sucede cuando un usuario falsifica su identidad para calificar de manera negativa o positiva, esto sesga la popularidad del artículo y reduce la confiabilidad del RS.
  Posible solución: Implementar técnicas de prevención para detectar a los atacantes analizando su tiempo de interacción, ya que suele ser muy rápida y realizar un promedio de las calificaciones de los artículos, esto permite identificar calificaciones muy altas o bajas, si las demás son neutras. Como paso final, eliminar las calificaciones y los perfiles falsos.

- **Sinonimia:** Ocurre cuando el nombre de varios artículos se componen de palabras similares pero su uso es diferente. Por ejemplo, mesa de noche y mesa gamer. Esto disminuye la precisión de las recomendaciones, ya que un RS no tiene la capacidad de distinguir automáticamente la diferencia.
  Posible solución: Implementar técnicas como filtrado demográfico, expansión de consultas o descomposición en valores singulares (*Singular Value Decomposition, SVD*).

- **Latencia:** Se produce cuando se incorporan nuevos usuarios o artículos frecuentemente y se recalcula la medida de similitud, esto aumenta el proceso computacional y el RS tiene problemas para recomendar nuevos artículos.
  Posible solución: Usar filtrado basado en contenido o realizar los cálculos en un entorno sin conexión o también emplear técnicas de clustering mejora el rendimiento.

- **Escasez:** Se produce cuando los usuarios activos califican muy pocos artículos, generando dispersión en los datos y falta de valores esperados, esto disminuye la precisión de la recomendación.
  Posible solución: Implementar técnicas como filtrado demográfico, enfoques basados en modelos o SVD.

- **Ovejas grises:** Sucede cuando un usuario no tiene similitud con otro (s) y presenta problemas para recomendar artículos relevantes.
  Posible solución: Usar filtrado colaborativo basado en artículos, basado en contenido o popularidad.

- **Escalabilidad:** Se produce cuando la cantidad de usuarios y/o artículos, aumenta de forma masiva. Esto incrementa el proceso computacional para el entrenamiento y el cálculo de las recomendaciones.
  Posible solución: Implementar técnicas como reducción de dimensionalidad y/o clustering.

2.5.10. Consideraciones para los RS

Para desarrollar un RS se necesita tener un contexto del sector al cual se le aplica, esto permite saber cómo manipular la información y que problemas controlar. En estos sistemas se requiere saber tipo de recomendación que se quiera realizar (general o personalizada), el filtro más adecuado, las similitudes y las métricas de evaluación que aseguren su correcto desempeño. El estudio realizado por Roy y Dutta (2022) concluye que los lenguajes de programación más comunes para el desarrollo de los RS son Java y Python, por su disponibilidad de bibliotecas que facilitan su creación.

2.5.11. Propuesta de implementación de técnicas de clusterización para la recomendación personalizada de productos para una plataforma de e-commerce en Honduras

Un proyecto realizado por Brizuela y Erazo (2025) en la Universidad Tecnológica Centroamericana *(UNITEC)* implementó técnicas de clusterización para realizar recomendaciones personalizadas para datos implícitos, a tiendas de productos de ferretería y maquillaje en la plataforma LynxPay de comercio electrónico en Honduras.

Recolección de datos

Se extrajeron los datos de los años 2021 al 2024 a través de consultas SQL usando DBeaver y PuTTY. Tomaron transacciones de datos donde los clientes tuvieran al menos una compra en productos del carrito, montos comprados, fechas y datos sociodemográficos.

Preprocesamiento de datos

La implementación del algoritmo se hizo en Python usando ETL. Generaron datos como tendencia de compra a través de las fechas y el gasto promedio. Crearon variables usando el modelo RFM y eliminaron valores nulos, mientras que para las variables demográficas se aplicó la imputación con la moda.

Implementación del modelo

1. Clustering usando K-means y DBSCAN. Al realizar DBSCAN encontraron que su coeficiente de silueta (0.19) no era bueno y lo descartaron, se usó solo K-means que generó 3 clústeres. El clúster más grande tiene el 76.57 % de los clientes, corresponde a los clientes promedios que realizan compras para el hogar o profesionales en construcción. El siguiente clúster tiene el 14.39 %, es un perfil más diversificado y no siempre se compra el mismo producto. Finalmente, el tercer clúster con 9.04 %, son los cazadores de ofertas ya que compran en liquidación u ofertas.

2. El siguiente paso fue implementar un HRS, usaron filtrado colaborativo y basado en contenido para generar recomendaciones dentro de cada clúster.

Resultados

Usaron diferentes métricas de evaluación para medir el rendimiento del algoritmo de clustering y del HRS.

- El mejor algoritmo de clustering fue el K-means con un coeficiente de silueta = 0.58, Calinski–Harabasz = 282 y Davies–Bouldin = 0.86.
- Para HRS trabajaron con Precision@10 en ferretería y maquillaje, alcanzaron en promedio un 4.52 % y 8.37 % de aciertos en las recomendaciones respectivamente.

Limitaciones

Tuvieron problemas de inicio en frío y se dificultó la recomendación a usuarios nuevos o con pocas compras.

Conclusiones

- El enfoque híbrido mejoró la precisión de las recomendaciones en comparación con un RS de solo filtrado colaborativo o general.
- El tipo de negocio al que se le implementa un PRS es importante, pues la precisión del modelo varía. Mostró que para este estudio las recomendaciones fueron más precisas en maquillaje.
- El sistema fue considerado viable a nivel económico y técnico. Confirmaron que la clusterización mejora la personalización de las recomendaciones y aumenta las ventas de la página web a medida que se generan más recomendaciones a los clientes. Recomiendan reentrenarlo al menos dos veces al año para tener recomendaciones precisas.

2.6. Relación entre clustering y RS

Los RS basados en clustering ofrecen un enfoque escalable al limitar el espacio de búsqueda a clústeres específicos y a su vez facilitan la interpretación dentro de cada segmento, útil para comprender e interpretar las recomendaciones que se realizan. También ayudan a reducir el problema de inicio en frío, al momento de integrarse un nuevo artículo o usuario el sistema de recomendación los puede asociar con grupos ya existentes (GeeksforGeeks, 2024). Además, al trabajar dentro de clústeres homogéneos, se reduce el ruido y aumenta la precisión de las recomendaciones porque las preferencias de los usuarios en cada grupo tienden a ser más estables (Aggarwal, 2016). Presenta desafíos como la selección adecuada del número de clústeres y el manejo de datos dinámicos, sin embargo, es una herramienta que ha demostrado ser de gran valor para revelar los patrones ocultos y diferenciarse con los RS tradicionales al agrupar puntos de datos en función de similitudes, reduce la complejidad computacional y sirve como preprocesador para el posterior uso de un filtrado (Meegle, 2025).

Capítulo 3

Metodología

En este capítulo se presenta la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*), esta permite desarrollar el trabajo a través de las siguientes etapas: comprensión de los datos, preparación, modelado y evaluación (Data Science PM, 2024). Adicionalmente, se siguen las pautas establecidas por Sampieri *et al.* (2003) para describir cómo fue llevada a cabo la investigación, que incluye: enfoque, alcance, contexto de la investigación, casos, universo y muestra, diseño utilizado y procedimiento.

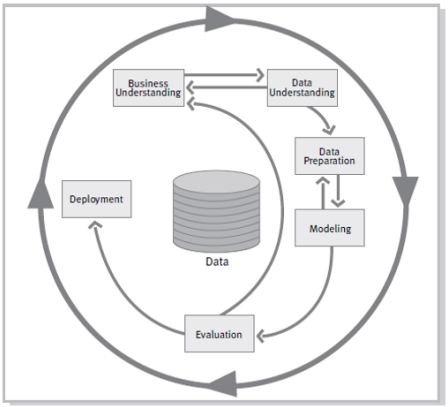

Figura 3.1: Metodología CRISP-DM

3.1. Enfoque

Este trabajo utiliza una metodología cuantitativa ya que se basa en aplicar modelos matemáticos y estadísticos para tomar decisiones provenientes del historial de ventas y de las fichas técnicas de los artículos. Este enfoque permite encontrar una técnica de clustering que segmente los comportamientos en compra a través del modelo RFM, mida las relaciones de las variables y de paso a la construcción de un PRS, al cual se le evalúa su desempeño mediante métricas.

3.2. Alcance

Para comprender el negocio es importante tener en cuenta que se cuenta con información transaccional y descriptiva de los artículos, por lo tanto, este trabajo tiene un alcance exploratorio y correlacional. En la fase exploratoria se crea un modelo RFM y se le aplican técnicas de clustering con el objetivo de identificar segmentos de clientes con patrones de compra similares. En la etapa correlacional, se analiza la relación entre las variables del modelo RFM, con el objetivo de identificar las que más aportan al momento de interpretar los comportamientos de compra y así generar PRS coherentes y correctos. Dentro de cada clúster se implementa un PRS teniendo en cuenta el comportamiento de cada grupo y la información descriptiva de los artículos. Es importante tener en cuenta que el alcance de este estudio va hasta la etapa de evaluación de los PRS, siguiendo la metodología CRISP-DM.

3.3. Contexto de la investigación

Teniendo en cuenta el objetivo del estudio presentado en el capítulo de introducción, se tienen datos transaccionales desde enero de 2018 hasta el 31 de diciembre de 2024, se cuenta con las ventas históricas y las fichas técnicas de las fragancias. Esta información fue extraída por el área de contabilidad de los sistemas World Office y Avancy, en formatos .xlsx y .csv.

- Software World Office, cuenta con los registros correspondientes a las ventas del período 2021-2024 y las fichas técnicas de los artículos.
- Avancy, cuenta con los registros históricos correspondientes a las ventas del período 2018-2020.

3.4. Casos, universo y muestra

- **Casos:** Cada caso corresponde a una transacción de venta de artículos de la línea de fragancias en una fecha específica registrada en las bases de datos. Para cada caso en el historial de ventas se consideran variables como: Cliente, documento, vendedor, factura, fecha emisión, descripción, unidad de medida, cantidad, grupo, subgrupo, unitario, descuento, IVA, total y ciudad.
  Y para las fichas técnicas se cuentan con variables como: Nombre comercial, código, lote no., cantidad (kg), fecha de fabricación, fecha de vencimiento, estado físico, color, perfil olfativo, densidad (g/ml) (20 °C), índice de refracción (20 °C) e inspiración.
- **Universo:** Todos los usuarios que realizaron compras en la línea de fragancias durante el periodo 2018-2024.
- **Muestra:** Se tomó la decisión de trabajar con todos los registros identificados de fragancias. Esto asegura que todas las características de la base de datos sean consideradas, dando un panorama más completo de los comportamientos, con el objetivo de obtener

resultados reales y válidos.

Justificación:

- Se tiene acceso completo a la base de datos, se registran 1240 usuarios y 520 artículos únicos, estos son registros transaccionales del período 2018-2020. Permitiendo realizar un análisis de toda la población.
- Se tienen 74,100 registros, manejables para trabajar con herramientas y código en Python, facilitando el proceso computacional.
- Los registros disponibles son completos y verificables, con información estructurada sobre cada variable y transacciones realizadas.

3.5. Variables del estudio

Se comprendió la estructura de los archivos, el tipo de variables, la presencia de valores atípicos, las distribuciones y las correlaciones. Con el fin de identificar las transformaciones y variables necesarias para realizar el modelo RFM.

3.5.1. Historial de ventas

Los datos de ventas se encontraban en formato .xlsx, con hojas agrupadas que dificultaban la automatización del procesamiento. A continuación, se identifican las variables y sus valores nulos.

Tipos de variables

El conjunto de datos está compuesto por 17 variables y 74,100 registros. De estas variables, 2 son categóricas, 7 son numéricas, 1 es fecha y 7 son cadenas de caracteres.

Como se observa en el cuadro 3.1, la variable *Descripción* contiene el código y el nombre del producto, la variable *Cliente* presenta errores de digitación. Por motivos de seguridad, el *NIT* se debe cambia por un valor ficticio. Las variables *UM* y *Cantidad* están cruzadas. *Documento* tiene la información de las facturas. Los valores faltantes de *Unitario*, *Total*, *Precio_promedio*, *Descuento*, *Neto* e *IVA* se pueden completar manualmente con fórmulas básicas.

Cuadro 3.1: Estructura inicial del histórico de ventas

Variable	**Tipo de dato**	**Definición operacional**	**Valores nulos**
Cliente	Texto	Nombre del cliente (persona natural o jurídica) que realiza la compra.	11
NIT	Numérico (categórico)	Código único de identificación del cliente.	0
Vendedor	Texto	Nombre del vendedor que realizó la venta.	13,140
Documento	Numérico (categórico)	Número asignado a la factura de venta.	0
Fecha	Fecha	Fecha que incluye día/mes/año de la venta.	0
Descripción	Texto	Nombre del artículo vendido seguido del código que lo identifica.	0
Grupo	Texto	Clasificación comercial dentro del catálogo.	50,285
Sub_Grupo	Texto	Clasificación específica del grupo comercial.	50,285
UM	Numérico	Cantidad de artículo vendido.	0
Cantidad	Texto	Unidad de medida abreviada para los kilogramos (KG)	0
Unitario	Numérico	Valor monetario individual del artículo antes de impuestos.	24
Total	Numérico	Valor total de la venta.	24
Ciudad	Texto	Ciudad del cliente desde donde compra.	50,285
Precio_promedio	Numérico	Precio promedio efectivo de venta.	24
Descuento	Numérico	Monto monetario descontado al total de la venta.	30,565
Neto	Numérico	Monto antes de IVA.	30,565
IVA	Numérico	Impuesto al valor agregado aplicado sobre el monto neto.	30,565

3.5.2. Fichas técnicas

Las fichas técnicas se encontraban en formato .csv. El conjunto de datos está compuesto por 12 variables y 598 fragancias. De estas variables, 2 son categóricas, 3 son numéricas, 2 son fechas y 5 son cadenas de caracteres.

En el cuadro 3.2 se observa que las variables útiles para el PRS son el *Nombre comercial*, *Código*, *Perfil olfativo* e *Inspiración*. Los detalles químicos y las fechas no aportan a la personalización de las fragancias, ya que los usuarios no se fijan en esos detalles, sino en notas olfativas, marcas o el uso industrial o personal que se le dé al producto.

Tipos de variables

Cuadro 3.2: Estructura inicial de las fichas técnicas

Variable	Tipo de dato	Definición operacional	Valores nulos
Nombre comercial	Texto	Nombre del producto.	0
Código	Numérico (categórico)	Identificador único del producto.	0
Lote No.	Numérico (categórico)	Identificador del lote de producción.	0
Cantidad (kg)	Numérico	Cantidad de artículos fabricados del lote	0
Fecha de fabricación	Fecha	Fecha en la que se fabricó el lote	0
Fecha de vencimiento	Fecha	Fecha en la que se expira el lote	0
Estado físico	Texto	Descripción del aspecto físico (líquido, homogéneo).	0
Color	Texto	Color visual del producto.	0
Perfil olfativo	Texto	Descripción sensorial del artículo (familia olfativa).	0
Densidad (g/ml) (20°C)	Numérico	Valor masa/volumen reportado en la ficha técnica.	0
Índice de refracción (20°C)	Numérico	Valor de refracción de la luz reportado en la ficha técnica.	0
Inspiración	Texto	Fragancia o marca que inspira la fórmula.	0

Una vez identificadas las transformaciones que se deben realizar a ambos datasets, se procede con la limpieza de los datos, asegurando la integridad en la información para realizar modelos de calidad.

3.5.3. Selección y limpieza

En ambos datasets se eliminan caracteres especiales de las variables, se quitan mayúsculas, tildes y espacios. Esto permite manipular la información más fácil y eliminar registros duplicados.

Historial de ventas

De la variable *Descripción* se crean dos variables: *codigo* y *producto*. Los registros nulos de la variable *Cliente* se eliminan porque no permiten realizar PRS. El contenido de *NIT* se reemplaza por un valor aleatorio y se guarda en la variable de *cliente_id*. Se intercambia el contenido de las variables *UM* y *Cantidad*. Se cambia el nombre de *Documento* a *factura*. Se extrae el año de *fecha* y se almacena en la variable llamada *anio*.
Los valores faltantes se completaron con las siguientes fórmulas:
$\text{unitario} = \frac{\text{total}}{\text{cantidad}}$
$\text{total} = \text{cantidad} \cdot \text{unitario}$
Teniendo en cuenta el año de la transacción se busca el IVA correspondiente para completar la información: $\text{neto} = \frac{\text{total}}{\text{IVA}}$
Con esta información completa, se calcula: $\text{iva} = \text{total} - \text{neto}$

Se crea la variable iva_pct para observar el porcentaje en cada registro: iva_pct = $\frac{\text{iva}}{\text{neta}}$
La información de *vendedor*, *ciudad*, *grupo* y *sub_grupo* se completó con el historial teniendo en cuenta la agrupación por *cliente_id*.
Se eliminan las variables *descuento* y *precio_promedio* ya que no dependen para el cálculo de otra variable ni para el modelo RFM.
El conjunto de datos se compone de 73,089 registros con 18 columnas después completar la información de las variables.

Cuadro 3.3: Estructura final del conjunto de datos del histórico de ventas

Variable	**Tipo de dato**	**Valores nulos**
cliente	object	0
cliente_id	int	0
vendedor	object	0
factura	int	0
fecha	datetime64[ns]	0
anio	int	0
codigo	int	0
producto	object	0
grupo	object	0
sub_grupo	object	0
UM	object	0
cantidad	float64	0
unitario	float64	0
total	float64	0
ciudad	object	43,962
neto	float64	0
iva	float64	0
iva_pct	float64	0

Se evalúa la proporción de valores faltantes en la variable *ciudad* con la siguiente fórmula: $\frac{\textbf{Total de registros nulos}}{\textbf{Total de registros en el dataset}}$. Para los valores faltantes se realiza imputación por la moda, basándose en el vendedor que realizó la venta. Para detectar la presencia de valores atípicos, se realizan gráficos de caja (*boxplots*) y se calcula el porcentaje de cada variable con el rango intercuartílico $IQR = Q_3 - Q_1$. Para analizar la distribución de los datos se realizan gráficos Q-Q plot, histogramas y una prueba de normalidad de Kolmogorov–Smirnov (KS) asumiendo un error del 5 %, para el análisis estadístico correspondiente. Se observa la relación de las variables a través de la matriz de correlación con el parámetro que corresponda una vez se sepa la distribución de las variables. Finalmente, se realiza análisis descriptivo de los usuarios, artículos y ciudades para entender el comportamiento de compra.

Fichas técnicas

Se eliminan las variables que no sirven para realizar el PRS. Se separa el *perfil_olfativo* en 5 columnas, las cuales contienen una sola palabra descriptiva. Se crea una nueva variable llamada *marca* la cual se deriva de *inspiración*, se crean las columnas *grupo* y *sub_grupo*, las cuales se extrajeron del dataset del historial de ventas unidad por *codigo*.

Cuadro 3.4: Estructura final del conjunto de datos de las fichas técnicas

Variable	Tipo de dato	Valores nulos
codigo	int	0
producto	object	0
grupo	object	0
sub_grupo	object	0
color	object	0
perfil_olfativo_1	object	0
perfil_olfativo_2	object	0
perfil_olfativo_3	object	0
perfil_olfativo_4	object	0
perfil_olfativo_5	object	0
marca	object	0
inspiracion	object	0

Se observa que las variables no cuentan con valores nulos y están listas para ser codificadas para hacer PRS.

Modelo RFM

Teniendo en cuenta el modelo tradicional de las variables RFM expuesto en el marco teórico, se decide agregar más variables basadas en el estudio de Chavhan *et al.* (2025), para ensayar si se obtienen mejores resultados para el clustering e identificar los perfiles de los usuarios.
En el siguiente cuadro se describe la construcción de cada variable:

Cuadro 3.5: Construcción del modelo RFM + variables

Componente	Variables utilizadas	Cálculo
Recencia (R)	cliente_id, fecha	Resta de la fecha máxima del conjunto de datos (30/12/2024) menos la última fecha de transacción de cada *cliente_id*.
Frecuencia (F)	cliente_id, factura	Cuenta del número de facturas únicas de cada *cliente_id*.
Monetario (M)	cliente_id, total	Suma de *total* agrupada por *cliente_id*.
Cantidad total	cliente_id, cantidad	Suma de *cantidad* agrupada por *cliente_id*.
Número de productos	cliente_id, codigo	Cuenta del número de códigos únicos de cada *cliente_id*.
Unión final	cliente_id	Estas variables se unieron a través de *Cliente_id*. Para formar otras tres variables.
Compra promedio	cliente_id, fecha	Días (tiempo medio) que pasan entre las fechas de compra de cada *cliente_id*.
Ticket promedio	monetario, frecuencia	División de *monetario* entre *frecuencia* para obtener el gasto medio por factura.
% compras fin de semana	cliente_id, fecha	Porcentaje de las ventas en sábado/domingo para cada *cliente_id*.

Se deben eliminar los registros donde se identifiquen usuarios con menos de tres compras en la variable *Número de productos*, ya que para el PRS se complica realizar predicciones a usuarios con un historial pequeño.
Se realiza análisis descriptivo de las variables creadas para entender el comportamiento, se grafican las distribuciones y se mide la relación entre ellas a través de la matriz de correlación

con el parámetro que corresponda una vez se sepa la distribución de las variables.
Siguiendo los estudios analizados en el cuadro 2.1 se sabe que trabajar con el modelo RFM requiere transformar y normalizar las variables para garantizar buenos clústeres, se prueba con iteraciones de logaritmo natural solo y más uno, Box-Cox, Min-Max y Z-score. Se evalúa el ajuste de esas transformaciones a través de la asimetría (*skewness*) y curtosis (*kurtosis*), ya que permiten identificar patrones y tomar decisiones informadas en el análisis estadístico (Sulca, 2018).
Se trabaja con la librería de Python Scipy (GeeksforGeeks, 2022) que permite calcular estos ajustes, siguen la fórmula:

$$\text{Skewness}(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right]$$

$$\text{Kurtosis}(X) = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right]$$

En caso de contar con variables RFM con alta correlación, se sigue el estudio de Darraz et al. (2025) donde decidieron hacer PCA para reducir la dimensionalidad y realizar clustering con esas dimensiones identificadas, en este trabajo se va a priorizar una varianza acumulada entre 0.7 y 0.9.

3.6. Diseño

La investigación es de tipo no experimental, transeccional correlacional. Teniendo en cuenta que no se manipulan las variables, sino que se observan en su contexto natural, se analizan en el entorno real sin intervenciones por parte del investigador. Además es transeccional, ya que la recolección de los datos se realiza en un único momento en el tiempo, requiriendo una medición de datos en un punto específico y correlacional porque se busca identificar la relación entre las variables RFM, los segmentos construidos y el desempeño de los PRS.
Brizuela y Erazo (2025) realizaron una matriz metodológica que ayuda a organizar el proceso elaborativo e identificar las variables claves para las segmentaciones que se pueden realizar y llegar a una buena predicción en los PRS. Para ambos modelos se realiza un proceso iterativo, donde se evalúan múltiples combinaciones hasta encontrar los mejores resultados.

- En clustering se aplican diferentes transformaciones de distribución, métodos de escalado, uso de PCA con diferentes varianzas acumuladas y sin aplicar. Para los algoritmos se prueba con diferentes valores en los hiperparámetros.
  A continuación, se muestra en la figura 3.2 un resumen del proceso para la segmentación de los comportamientos de los usuarios:

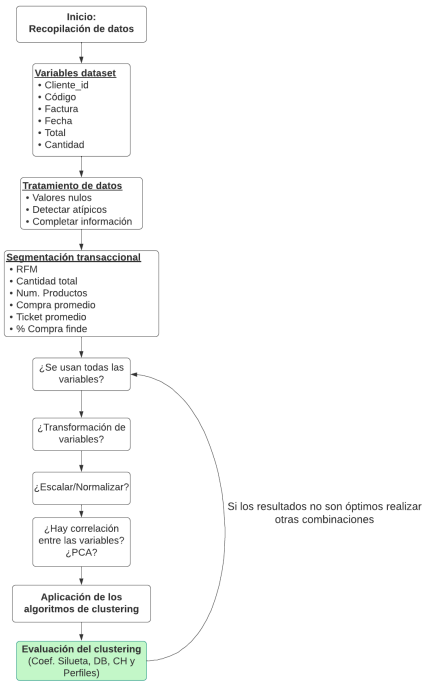

Figura 3.2: Procedimiento del clustering

La segmentación transaccional se basa en el análisis de datos relacionados con las transacciones realizadas por los usuarios, se muestran variables como la cantidad, los artículos y el monto total gastado. Son muy utilizadas en el área de marketing, permiten comprender la interacción de los usuarios con el negocio (Parsons, 2024). A partir de esta segmentación se puede crear el modelo RFM de manera eficiente, permitiendo comprender rápidamente la información y reducir la dimensionalidad de los datos (Chandana,2024).

- En PRS se aplican diferentes combinaciones de matrices de utilidad, filtrados, similitudes y normalizaciones para los puntajes. Se itera con diferentes valores para filtrar el ruido de popularidad y peso en α. Para evaluar el desempeño de las recomendaciones se trabaja con diferentes pliegues para la validación cruzada y valores K.
  Después de tener los clústeres formados, se procede a realizar el PRS. En la siguiente figura se presenta el resumen de los pasos a seguir:



Figura 3.3: Procedimiento de los PRS

3.7. Herramientas

Para el desarrollo de este trabajo se usaron diferentes herramientas que garantizan el correcto uso e integración de los datos, se utilizó el lenguaje de programación Python ya que maneja grandes volúmenes de datos y cuenta con una gran variedad de librerías especializadas en el análisis de datos (Roy y Dutta, 2022). Este lenguaje de programación se manejó dentro del entorno de Visual Studio Code. Esta herramienta permitió realizar las fases de la metodología CRISP-DM hasta la evaluación de los PRS implementados.
Las librerías que se usaron son tomadas de Python Documentation (s.f.):

- Pandas: Se usa para manipular y analizar datos estructurados. En el trabajo se usa para todo el preprocesamiento de los datos y fases del modelo CRISP-DM hasta la evaluación de los PRS.
- Numpy: Contiene cálculo numérico y manejo de grandes cantidades de datos. Se usa para estructurar los datos, crear vectores y calcular las distancias de los clústeres.
- Matplotlib: Genera la visualización de los datos. Se usa para todas las gráficas del trabajo.
- Seaborn: Se usa para la matriz de correlación con el mapa de calor y los histogramas.
- Scipy: Contiene herramientas para estadística avanzada, en el trabajo se usa para calcular las correlaciones entre las variables, aplicar la transformación box-cox, medir la asimetría y curtosis de los datos después de transformarlos.

- Rapidfuzz y Thefuzz: Detectan y unifican registros de artículos y clientes con nombres parecidos. Se usa para encontrar nombres con características similares y emparejarlos para que no genere registros duplicados por errores de digitación.
- Sklearn.neighbors: Encuentra los puntos de clientes o artículos más cercanos, con base en una métrica de distancia. Se usa para calcular el gráfico k-distances antes de aplicar DBSCAN.
- Sklearn.cluster: Se usa para la agrupación de datos sin etiqueta, en el trabajo se usa para realizar el clustering de los algoritmos como K-means, DBSCAN, jerárquico y BIRCH.
- Sklearn.preprocessing: Se usa para escalar y transformar los datos. En el trabajo se usa para el manejo de valores atípicos como RobustScaler, normalización Z-score llamada StandardScaler y para que el vector tenga longitud igual a 1 en los usuarios.
- Sklearn.mixture: Se usa para modelos probabilísticos, en el trabajo se usa para crear el modelo de clustering GMM.
- Sklearn.metrics: Se usa para evaluar la calidad de los modelos, en el trabajo se usa para las métricas del coeficiente de silueta, DB y CH. Con pairwise se usa para calcular la similitud coseno entre vectores.
- Sklearn.decomposition: Se usa para reducir la dimensionalidad de los datos, en el trabajo se usa para realizar PCA de las variables.
- Sklearn.model_selection: Se usa para dividir el conjunto de datos en k subconjuntos, en el trabajo se usa para validación cruzada.
- Fcmeans: Se usa para crear el modelo de Fuzzy C-means. Permite que un mismo dato pertenezca a varios clústeres con distintos porcentajes de certeza.

3.8. Procedimiento

A continuación, se describe el procedimiento para construir y evaluar los modelos de clustering y PRS.

3.8.1. Clustering

Cada configuración fue procesada a través de diversos algoritmos de clustering como K-means, GMM, BIRCH, jerárquico, DBSCAN y Fuzzy C-means seleccionados por su efectividad en los estudios del cuadro 2.1 del marco teórico, ajustando sus respectivos umbrales e hiperparámetros.
Teniendo en cuenta la siguiente tabla obtenida con ScienceDirect AI (2026) se guía el trabajo para escoger el coeficiente de silueta adecuado, considerando los rangos y las literaturas académicas a continuación:

Cuadro 3.6: Rangos y literatura del coeficiente de silueta, ScienceDirect AI (2026)

Rango del coef. Silueta	**Interpretación**	**Referencias**
$> 0{,}7$	Agrupamiento muy robusto y bien definido	Ballestar *et al.*, 2022; Son *et al.*, 2025
$0{,}5 - 0{,}7$	Buena calidad de agrupamiento	Ballestar *et al.*, 2024
$0{,}4 - 0{,}5$	Aceptable en la práctica, pero no óptimo	Deng *et al.*, 2021
$0 - 0{,}4$	Agrupamiento débil o poco definido	Zhang *et al.*, 2026; Ketan & Charles, 2020
< 0	Mala asignación, posible necesidad de revisión	Deng *et al.*, 2021; Zhang *et al.*, 2026; Ketan & Charles, 2020

Hiperparámetros de los algoritmos

Para obtener el mejor modelo segmentado, se configuraron los hiperparámetros en los algoritmos, teniendo en cuenta que el objetivo es obtener más de dos clústeres porque se va a poder realizar un PRS más preciso de acuerdo a los comportamientos de los usuarios, manejando $k \in [3, 10]$ y una semilla para todos los algoritmos. Este proceso de iteración permitió evidenciar los resultados a través del desempeño en las métricas de evaluación y en los perfiles de cada clúster:

- **K-means:** Se aplica inicialización *K-means++* para evitar que el algoritmo quede en mínimos locales, se ejecuta 100 veces con diferentes centroides aleatorios y un límite de 500 iteraciones por ejecución para que los centroides alcancen una convergencia absoluta. Se utiliza el algoritmo *elkan* para optimizar los cálculos de distancia ya que usa la desigualdad triangular y se utiliza la distancia euclidiana.

- **GMM:** Se prueban tres tipos de covarianza, *diag* asume que las variables son independientes, con menos sobreajuste. *tied* asume que todos tienen la misma matriz, obliga a que tengan la misma forma y *full* donde cada clúster tiene su matriz independiente. Se realizan 20 inicializaciones independientes para que el modelo sea robusto y un límite de 500 iteraciones por ejecución.

- **DBSCAN:** Se prueba con valores en épsilon (ϵ) $\in [0{,}1; 1{,}0]$, MinPts $\in \{5, 10, 15\}$) y la distancia euclidiana.

- **BIRCH:** Se prueban diferentes umbrales de radio $\in [0{,}3; 0{,}6]$ y un factor de ramificación igual a 50.

- **Jerárquico:** Se prueban con tres medidas de distancia euclídea, manhattan y coseno. Se utilizan dos enlaces, promedio y completo para calcular la distancia entre los puntos.

- **Fuzzy C-means:** Se prueba con 500 iteraciones, siguiendo el estudio de Santoyo *et al.* (2023) se define el grado de difusividad $m = 2$, el clúster donde el usuario tenga la probabilidad más grande se asigna ahí.

Métricas de evaluación

Para evaluar el rendimiento de los algoritmos de agrupación, en este trabajo se va a usar el coeficiente de silueta, los índices de Davies–Bouldin y Calinski-Harabasz, teniendo en cuenta que estas métricas son las más usadas en el cuadro 2.1 expuesto. Adicionalmente, es importante interpretar los perfiles que construya cada algoritmo, donde se evidencie una separación adecuada y coherente entre cada clúster, tal y como lo manejaron Chen *et al.* (2012). Teniendo en cuenta la información de Icesi Editorial (2025) se entiende el uso de cada métrica.

- **Coeficiente de silueta:** Calcula la similitud de cada individuo con el clúster en el que se encuentra, en comparación a los demás clústeres.

$$a(i) = \frac{1}{|C_i| - 1} \sum_{\substack{j \in C_i \\ j \neq i}} d(i, j)$$

$d(i, j)$: Distancia entre el individuo i e j que pertenecen al clúster C_i.
$a(i)$: Es la distancia promedio del individuo i a todos los otros puntos del clúster C_i.

Después se tiene a $b(i)$, una medida que determina qué tan cerca se encuentra el clúster más cercano al individuo i (el vecino).

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j)$$

C_k: Clúster distinto a C_i.
Finalmente se tiene la siguiente fórmula para calcular la silueta para el individuo i, se define como una combinación de $a(i)$ y $b(i)$:

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

Los valores de $s(i)$ van a estar entre -1 y 1, donde un valor igual o cercano a 1 indica mejor calidad de los clústeres, porque significa que el individuo está bien ubicado en el clúster asignado.

- **Índice de Davies–Bouldin *(DB)*:** Mide la similitud promedio entre clústeres, calculando la distancia promedio dentro y la distancia entre los centroides de cada uno.

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left(\frac{S_i + S_j}{M_{ij}} \right)$$

S_i: Dispersión promedio dentro del clúster i.
M_{ij}: Distancia entre los centroides de los clústeres i y j.
Un valor bajo (cercano a 0) de DB es el ideal porque indica que los clústeres son compactos y tienen buena separación de los demás.

- **Índice de Calinski-Harabasz *(CH)*:** También conocido como criterio de razón de varianza. Realiza una validación interna del clúster, mide la dispersión interna del clúster y la distancia entre los centroides.

$$CH = \frac{\frac{\sum_{i=1}^{K} n_i \ \|c_i - c\|^2}{K-1}}{\frac{\sum_{i=1}^{K} \sum_{j=1}^{n_i} \|d_j - c_i\|^2}{N-K}}$$

n_i: número de puntos en el i-ésimo grupo.
c_i: centroide del i-ésimo grupo.
c: centroide global.
N: número total de puntos de datos.
K: número de grupos.
d_j: j-ésimo punto dentro del grupo i.
Un valor de CH alto es el ideal porque significa que los puntos de datos estan más dispersos por fuera de los clústeres formados que dentro.

3.8.2. PRS

Después de identificar los clústeres se realiza la matriz usuario–artículo, se prueba con dos matrices con el fin de identificar la que mejor se ajuste al modelo implementado. La primera opción es una matriz binaria, con 1 en los artículos comprados por cada usuario, esta decisión se toma basándose en el estudio de Stylianou y Pantelidou (2025), ya que al trabajar con datos implícitos, manipular de esta manera la información ayuda a tener la diversidad del catálogo de artículos comprados por cliente y estabilidad al calcular las similitudes. La segunda opción es realizar una matriz con el número total de compras que ha realizado cada usuario en el artículo. Por ejemplo:

Usuario / Artículo	**Artículo 1**	**Artículo 2**	**Artículo 3**	**Artículo 4**
Usuario 1	10	7	2	0
Usuario 2	0	19	0	1

Cuadro 3.7: Matriz usuario-artículo con el total de artículos comprados

Para trabajar con las fichas técnicas se realiza una matriz binaria de cada artículo donde se le pone uno a las características a las que pertenezca, se muestra el siguiente ejemplo:

Artículo/ Característica	**Frutal**	**Dulce**	**Floral**	**Polímeros**
Artículo 1	1	0	0	0
Artículo 2	0	1	1	0

Cuadro 3.8: Matriz artículo-característica binaria

Entrenamiento por clúster

Teniendo en cuenta que el comportamiento de compra cambia dependiendo del clúster que se esté trabajando, se realiza un PRS independiente para cada uno.
Una vez creada la matriz usuario–artículo y matriz de artículos, se unen a través del codigo del artículo para unificar la base. y realizar los siguientes pasos:

1. **Filtrar artículos populares:** El estudio de Klimashevskaia *et al.* (2024) demuestra que los artículos populares (más vendidos) suelen recibir más atención del algoritmo, perjudicando el objetivo de recomendar artículos diversos. Las soluciones que exponen para reducir el sesgo son (1) ajustar los algoritmos para equilibrar la popularidad y la diversidad de los artículos. (2) Hiperparámetros dinámicos ajustados de acuerdo al contexto. (3) Diversificar recomendaciones que prioricen la inclusión de artículos menos populares.
   Por lo tanto, para este estudio se implementa un hiperparámetro dinámico por clúster $\in [0{,}1; 0{,}85]$ eliminando los artículos más comprados dentro de cada uno en el entrenamiento, con el objetivo de reducir el sesgo de popularidad. Se sigue el siguiente procedimiento:

$$Popular_i = \frac{\mathrm{u}_i}{\mathrm{u}_T}$$

   u_i : Usuarios que compraron el artículo i.
   u_T : Usuarios totales en el clúster.
   La eliminación de los artículos se hace donde la popularidad sea mayor al hiperparámetro dinámico seleccionado ($Popular_i >$ Hiperparámetro).

2. **Filtrar usuarios con pocas compras:** Este procedimiento se realizó en la parte de la construcción del modelo RFM para garantizar que el PRS tenga suficiente información sobre los artículos comprados por cada usuario con un mínimo de 3 compras.

3. **Normalizar los usuarios:** A través de la matriz usuario-artículo se normaliza el vector de los usuarios. Evita que los usuarios que más compran dominen al momento de calcular la similitud. La idea es que tenga longitud 1.

$$\text{Usuarios} : \hat{u}_{o,i} = \frac{u_{o,i}}{\sqrt{\sum_{i=1}^{n} u_{o,i}^2}}$$

   $\hat{u}_{o,i}$: Valor del usuario objetivo o para el artículo i.
   n: Número total de artículos en el catálogo.

4. **Calcular similitudes**: Para saber numéricamente que tan parecidos son los usuarios y los artículos se calcula con la correlación de Pearson y la similitud del coseno.

5. **Normalizar/Estandarizar los puntajes**: Para tener los puntajes en la misma escala

se prueba con Min-Max y Z-score:

$$\text{Min-Max} = \frac{P_i - P_{min}}{P_{max} - P_{min}}, \quad \text{Min-Max} \in [0, 1]$$

$$\text{Z-score} = \frac{P_i - \mu}{\sigma}, \quad \text{Z-score} \in (-\infty, \infty)$$

6. **Diversificar las recomendaciones**: Para garantizar que los usuarios descubran artículos nuevos, se eliminan de la lista de recomendaciones los artículos que pertenezcan al historial de compras previas del usuario.

Filtrado basado en contenido

Para la creación del PRS basado en contenido, se trabaja con la matriz de artículos. Se calcula la similitud entre los artículos para crear una matriz artículo–artículo, esta contiene la similitud que comparten en sus descripciones de perfil olfativo, marca, uso, etc. Un valor igual a 1 o cercano indica que los artículos comparten las mismas características.

Artículo / Artículo	**Artículo 1**	**Artículo 2**	**Artículo 3**	**Artículo 4**
Artículo 1	1	0	0.01	0.3
Artículo 2	0	1	0.6	0

Cuadro 3.9: Matriz artículo–artículo con similitudes

UBCF

Para la creación del PRS basado en usuarios, se trabaja con la matriz usuario–artículo. Se calcula la similitud entre los usuarios (filas de la matriz) y se crea una matriz usuario–usuario, esta contiene la similitud entre los usuarios en el comportamiento de compra. Un valor igual a 1 o cercano indica que los usuarios tienen el mismo comportamiento de compra.

Usuario / Usuario	**Usuario 1**	**Usuario 2**	**Usuario 3**	**Usuario 4**
Usuario 1	1	0.5	0	0
Usuario 2	0.5	1	0	0.2

Cuadro 3.10: Matriz usuario–usuario con similitudes

IBCF

Para la creación del PRS basado en artículos, se trabaja con la matriz usuario–artículo transpuesta. Se calcula la similitud entre los artículos (filas de la matriz) y se crea una matriz artículo–artículo, esta contiene la similitud entre los artículos en el comportamiento de compra de los usuarios. Un valor igual a 1 o cercano indica que los artículos se compran al mismo tiempo por los usuarios.

Artículo / Artículo	Artículo 1	Artículo 2	Artículo 3	Artículo 4
Artículo 1	1	0.7	0.2	0
Artículo 2	0.7	1	0	0.1

Cuadro 3.11: Matriz artículo–artículo de U-A transpuesta con similitudes

HRS

Para la creación del PRS híbrido se guio el trabajo en la hibridación ponderada, Isinkaye *et al.* (2015) expusieron un trabajo donde combinaron los resultados de diferentes recomendadores a través de una fórmula lineal que integra IBCF y UBCF, ajustando los pesos de cada recomendador de acuerdo con los mejores resultados en las predicciones. En este trabajo se va a probar con la combinación de dos PRS, los cuales pueden ser UBCF, IBCF o contenido.

Recomendación popular

Se prueba con recomendaciones populares para evaluar el rendimiento en cada clúster, para estas recomendaciones se usa un hiperparámetro para el filtrado de ruido $> 0{,}80$ y así trabajar con artículos con alta frecuencia de compra. Para la creación de un RS popular se realiza una lista, contiene la sumatoria por artículo de la matriz usuario–artículo, donde cada artículo va acompañado del total de usuarios que lo compraron.
Para evitar que un artículo muy comprado se aleje mucho de los demás se aplica transformación logarítmica.

3.8.3. Puntaje de las recomendaciones

Para obtener el puntaje de recomendación de un artículo (A_i) se trabaja con la técnica del vecino más cercano explicada por Aggarwal (2016). El proceso consiste en calcular el producto punto entre el vector fila normalizado de los usuarios y la matriz de similitudes previamente calculadas.
- Para UBCF se tiene la siguiente fórmula:

$$\text{Puntaje}_{\text{UBCF}}(A_i) = \underbrace{\begin{bmatrix} s_{u,v_1} & s_{u,v_2} & \dots & s_{u,v_m} \end{bmatrix}}_{\text{Fila matriz usuario-usuario con similitudes}} \cdot \underbrace{\begin{bmatrix} \hat{u}_{v_1,i} \\ \hat{u}_{v_2,i} \\ \vdots \\ \hat{u}_{v_m,i} \end{bmatrix}}_{\text{Vector normalizado de los vecinos}}$$

- Para IBCF se tiene la siguiente fórmula:

$$\text{Puntaje}_{\text{IBCF}}(A_i) = \underbrace{\begin{bmatrix} \hat{u}_{o,1} & \hat{u}_{o,2} & \dots & \hat{u}_{o,n} \end{bmatrix}}_{\text{Vector normalizado de los usuarios}} \cdot \underbrace{\begin{bmatrix} 1 & s_{1,2} & \dots \\ s_{2,1} & 1 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}}_{\text{Matriz artículo-artículo con similitudes}}$$

- Para el filtrado basado en contenido se tiene la siguiente fórmula:

$$\text{Puntaje}_{\text{Contenido}}(A_i) = \underbrace{\begin{bmatrix}\hat{u}_{o,1} & \hat{u}_{o,2} & \dots & \hat{u}_{o,n}\end{bmatrix}}_{\text{Vector normalizado de los usuarios}} \cdot \underbrace{\begin{bmatrix}1 & sc_{1,2} & \dots \\ sc_{2,1} & 1 & \dots \\ \vdots & \vdots & \ddots\end{bmatrix}}_{\text{Matriz artículo-artículo con similitudes de características}}$$

- Para popular se tiene la siguiente fórmula:

$$Puntaje_{Popular}(A_i) = \log(\sum_{u=1}^{m} \hat{u}_i)$$

- m: Número total de usuarios únicos.
- $s_{u,v}$: Similitud en la matriz de UBCF entre el usuario objetivo u con el usuario vecino v.
- $s_{i,j}$: Similitud en la matriz de IBCF entre el artículo objetivo i con el artículo vecino j.
- $sc_{i,j}$: Similitud en la matriz de contenido entre las características de los artículos i y j.
- $\hat{u}_i$: Conteo total de usuarios únicos que compraron el artículo i.

- Para filtrado híbrido se trabaja con tres fórmulas para generar las recomendaciones y quedarse con la que tenga mejores resultados:

$$\text{Puntaje}_{\text{Híbrido}}(A_i) = \alpha \cdot Puntaje_{IBCF} + (1-\alpha) \cdot Puntaje_{Contenido} \tag{1}$$

$$\text{Puntaje}_{\text{Híbrido}}(A_i) = \alpha \cdot Puntaje_{IBCF} + (1-\alpha) \cdot Puntaje_{UBCF} \tag{2}$$

Las fórmulas permiten manejar los tres tipos de PRS de manera que si $\alpha = 1$, se obtiene solo la recomendación de IBCF, si $\alpha = 0$ se obtiene solo la recomendación basada en el contenido (1) o basada en usuarios (2).

$$\text{Puntaje}_{\text{Híbrido}}(A_i) = \alpha \cdot Puntaje_{UBCF} + (1-\alpha) \cdot Puntaje_{Contenido} \tag{3}$$

En este caso (3), si $\alpha = 1$, se obtiene solo la recomendación de UBCF, si $\alpha = 0$ se obtiene solo la recomendación basada en el contenido. Finalmente, para todos los casos si $0 < \alpha < 1$ se obtiene la recomendación de la hibridación ponderada, por lo tanto, se probará con $\alpha \in [0, 1]$.

Para cada usuario se debe normalizar el puntaje de la recomendación porque los rangos pueden ser diferentes para cada uno debido a la diferencia en el número de artículos o el valor en las similitudes. Se busca tener valores en una misma escala usando la normalización/estandarización de los puntajes.

Validación cruzada

Se realiza validación cruzada (K-Fold) para evaluar el rendimiento de las recomendaciones. Pacheco (2024) explica que es una técnica de remuestreo para evaluar el rendimiento del modelo. Parte los datos en K partes (folds) y separa los datos en entrenamiento y prueba, teniendo en cuenta que cada observación va a ocupar ambos roles. Realiza el entrenamiento para *K - 1* datos iterándolos hasta que recorra todos los K, siempre dejando a un grupo fuera para calcular el error de predicción de ese grupo. Finalmente, el rendimiento se estima como el promedio de todos los errores de predicción obtenidos y a través de las metricas de evaluación.
Para este trabajo se empieza dividiendo a los usuarios de cada clúster en $K \in [5, 10]$ grupos, con el objetivo de que el modelo aprenda de patrones generales y no del comportamiento de un clúster o un usuario específico. El proceso se repite K veces para asegurar que todos los usuarios pasen por prueba.
Adicionalmente, dentro de cada división se excluye un elemento (*Leave-One-Out* o *Leave-Last-Item-Out*, *LOO*), la idea es ocultar el último artículo adquirido por el usuario e intentar recomendarlo, entrenando al modelo con los artículos $1, 2, \cdots, N-1$ y evaluar si logra predecir el artículo N. Es una técnica utilizada en estudios de Zan *et al.* (2021), Chen *et al.* (2021) y Rendle *et al.* (2012), útil para evitar el data leakage, reducir el sesgo en usuarios con pocas compras, ya que se estandariza al eliminar y buscar predecir la última compra realizada para todos. Finalmente, este proceso permite evaluar el rendimiento a través de métricas con Top-K como HR@K, NDCG@K y Recall@K.

3.8.4. Métricas de evaluación

A pesar de que no existe una única medida estándar para evaluar los RS, Roy y Dutta (2022) concluyeron que las métricas más utilizadas son Recall, Precision, F1-score y Accuracy. Por otro lado, Chen *et al.* (2024, 2021) y Brizuela y Erazo (2025) midieron los resultados a través del parámetro Top-K, con Recall@20, NDCG@20 y Precision@10 respectivamente.
Las métricas con Top-K son más especiales para los RS, busca recomendar artículos que aparezcan dentro de la lista de K. En la mayoría de estudios se trabaja con $K \in [5, 10, 20]$, ya que un usuario no consume todos los artículos del catálogo. Para este trabajo siguiendo la definición de Chen *et al.* (2024), se ejecuta el sistema con los valores de $K \in [5, 10, 20]$, con Recall@K y NDCG@K.

$$Recall@K = \frac{1}{|U|} \sum_{u \in U} \frac{|I(u) \cap I_{Rec-K}(u)|}{|I(u)|}$$

- $|U|$: Número total de usuarios evaluados
- $I(u)$: Artículo ocultado en LOO que el usuario u compró
- $I_{Rec-K}(u)$: K artículos recomendados por el sistema

Mide si el artículo oculto está dentro de la lista de las K recomendaciones, por lo tanto, para un usuario es 1 si aparece en el Top-K y 0 en otro caso. En conclusión, se busca un $Recall@K$

cercano a 1 porque indica que el modelo predice la siguiente compra de los usuarios.

$$DCG@K = \sum_{k=1}^{K} \frac{2^{\delta_u(k)} - 1}{\log_2(k+1)}$$

$$NDCG@K = \frac{1}{|U|} \sum_{u \in U} \frac{DCG@K}{IDCG@K}$$

- $\delta_u(k)$: Función binaria que indica si el artículo en la posición k es relevante
- $\log_2(k+1)$: Penaliza la posición del artículo, entre más alejado esté de los primeros puestos (k grande) más grande es el logaritmo y el resultado de la división es más pequeño
- $IDCG@K$: Valor máximo de $DCG@K$

Mide la calidad teniendo en cuenta el orden de las recomendaciones, si el artículo predicho queda en primer lugar $NDCG@K$ es igual a 1, entre más alejado esté del ranking, más bajo es el puntaje.
Adicionalmente, se calcula Recall ponderado para evaluar el desempeño del sistema general, teniendo en cuenta la cantidad de usuarios en cada clúster.

$$Recall@K_{ponderado} = \frac{\sum_{i=1}^{c} (Recall@K_i \times m_i)}{\sum_{i=1}^{c} m_i}$$

- c: Número de clústeres.
- $Recall@K_i$: Valor de Recall@K obtenido en el clúster i.
- m_i: Número de usuarios en el clúster i.

3.9. Consideraciones éticas y técnicas

El tratamiento de la información cumple con la Ley 1581 del 2012 sobre protección de datos personales en Colombia. Los datos se usaron exclusivamente con fines académicos y con acuerdo de confidencialidad. Se aplicaron medidas de anonimización para garantizar la seguridad de la información.

Capítulo 4

Resultados

En este capítulo se presentan los resultados obtenidos siguiendo la metodología del modelo CRISP-DM.

4.1. Variables del estudio

Para los datos del historial de ventas se analizan los valores nulos y atípicos. Se grafican las variables y se aplica una prueba a las distribuciones y correlaciones.

4.1.1. Valores nulos

La variable ciudad presenta una pérdida considerable de información con un porcentaje de valores nulos igual a 60.14 %, por lo tanto, se decide realizar imputación por la moda basándose en el vendedor que realizó la venta y si aún hay valores nulos se deja la ciudad con más ventas.

4.1.2. Valores atípicos

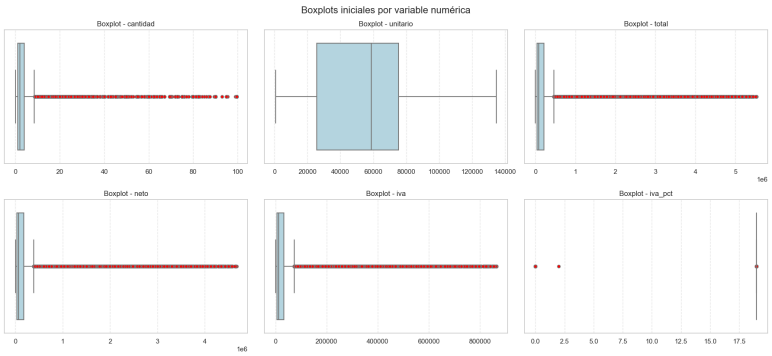

Figura 4.1: Gráfico boxplot para la detección de outliers en las variables numéricas

Se observa que las variables *cantidad*, *total*, *neto* e *iva* presentan outliers positivos, con valores más altos que el rango típico de la variable. Indicando que existen clientes o artículos con compras de valor o cantidad muy elevadas frente al promedio general. Es un comportamiento común en datos de ventas, ya que se pueden presentar por compras grandes o descuentos. Estos

resultados indican la necesidad de evaluar si son errores de registro, si deben eliminarse o transformarse.

Por otro lado, se observa que *iva_pct* tiene puntos que parecen ser errores de digitación, ya que el IVA en Colombia del año 2018 al 2024 siempre fue del 19 %.

Ahora se mide el porcentaje de outliers que se presentan en las variables respecto al dataset completo con IQR.

Cuadro 4.1: Número y porcentaje de outliers en las variables numéricas con IQR

Variable	**Outliers**	**Porcentaje (%)**
cantidad	10,724	14.6725
total	9,151	12.5203
neto	9,141	12.5066
iva	9,129	12.4902
iva_pct	7,180	9.8236
unitario	526	0.7196

Se observa que las variables *cantidad* (14.67 %) y *total* (12.52 %) presentan los mayores porcentajes de outliers, teniendo en cuenta que el modelo RFM se construye con total se debe evaluar la transformación de la variable para que no afecte los resultados. Por otro lado, *unitario* tiene el porcentaje de outliers más bajo.

4.1.3. Distribución

- Gráfico Q-Q

  A continuación, se realizó el grafico Q–Q que compara la distribución teórica normal y los valores observados de las variables. Se muestra *cantidad*, *unitario* y *total* por practicidad. Se observa que en los tres casos los datos no siguen la línea teórica, con curva hacia la derecha y nuevamente se evidencia la presencia de outliers, mostrando que no hay evidencia de que sigan una distribución normal.

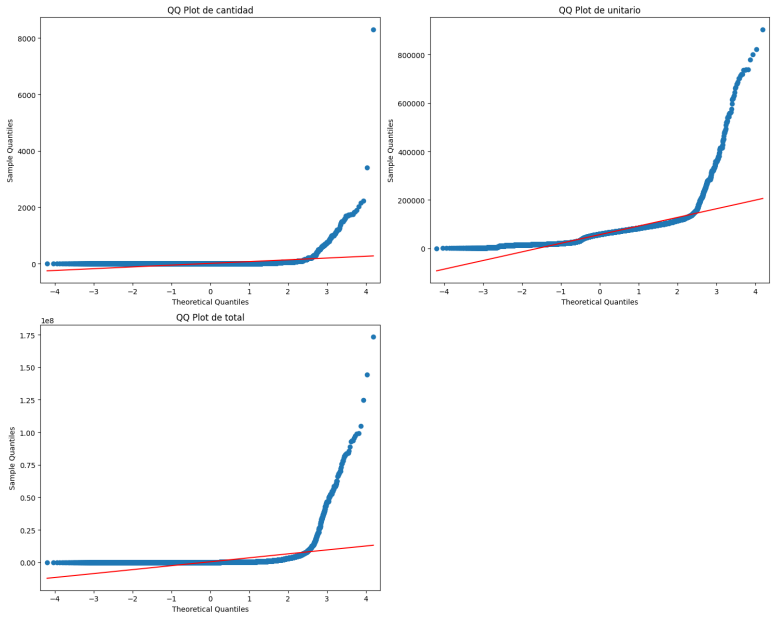

Figura 4.2: Gráfico QQ-plot

- Histogramas

  Se construyeron histogramas para observar la forma general de la distribución, identificar posibles sesgos y comparar con los resultados obtenidos del análisis Q-Q plot.

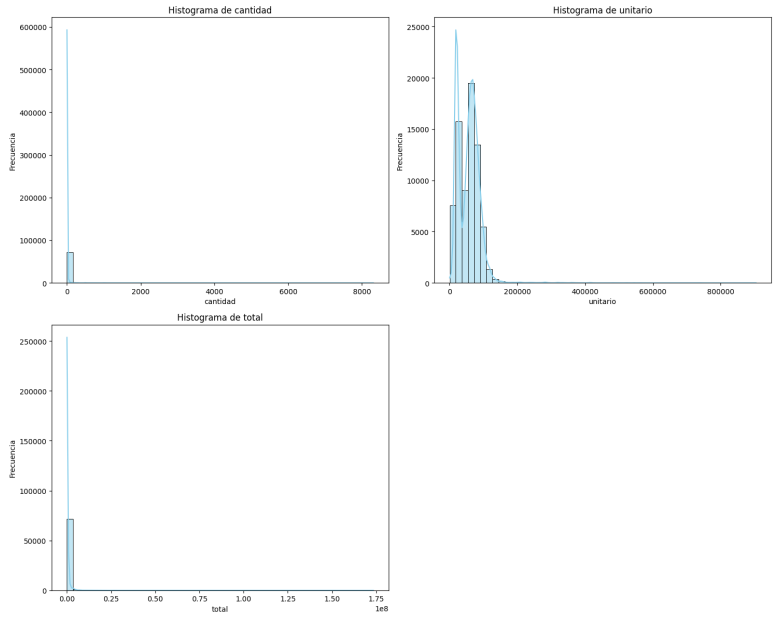

Figura 4.3: Histogramas

Se concluye que las gráficas indican que las variables no siguen una forma normal y que existen outliers que influyen en la media. Mostrando que la mayoría de las transacciones son pequeñas y unas pocas son muy grandes. Sin embargo, se realiza la prueba de Kolmogorov–Smirnov (KS):

$$\begin{cases} H_0 : \text{La muestra proviene de una distribución normal.} \\ H_1 : \text{La muestra no proviene de una distribución normal.} \end{cases}$$

Se obtuvo:

Cuadro 4.2: Prueba de normalidad de Kolmogorov–Smirnov para las variables numéricas

Variable	**Estadístico KS**	**p-valor**
cantidad	0.444	0.000
unitario	0.098	0.000
total	0.438	0.000
neto	0.438	0.000
iva	0.438	0.000
iva_pct	0.525	0.000

Con un p_valor = 0.000 en cada variable se rechaza la hipótesis nula (H_0). Por lo tanto, ninguna variable sigue una distribución normal.

Para el análisis estadístico es conveniente realizar transformaciones que permitan reducir el sesgo y obtener resultados confiables.

Matriz de correlación

Se realizó la matriz de correlación utilizando el coeficiente de Spearman, debido a que las variables presentan sesgos y valores atípicos.



Figura 4.4: Matriz de correlación

Se observa alta correlación positiva entre las variables *neto* e *iva* (0.99), lo cual indica que se comportan igual. *cantidad* presenta una fuerte correlación positiva con *total* (0.83), indicando que el valor total de venta depende principalmente de la cantidad de unidades vendidas. Por otro lado, *total*, *neto* e *iva* mantienen correlaciones altas (0.98 y 0.99).

4.1.4. Análisis descriptivo de los clientes y artículos

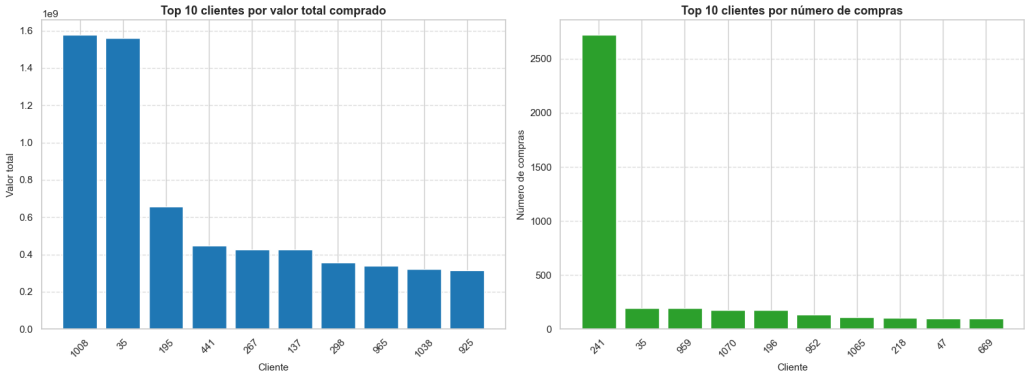

Figura 4.5: Top 10 distribución por cliente

Se observa que los clientes con id 1008 y 35 son los que más compran, superando los 10 mil millones, siendo los más importante en términos de ingresos.

En la frecuencia de compra, el cliente 241 realiza más transacciones (+2,500 compras) pero con montos pequeños. Sin embargo, este cliente se debe eliminar porque son registros de ventas a muchas personas y no se puede personalizar. Mientras que el cliente 35 se ubica en segundo lugar en número de compras, lo que indica que no compra con frecuencia, pero hace compras con montos grandes.

En la siguiente figura se visualiza el top 10 de artículos según el valor total comprado y el

número de compras. De las 598 fragancias presentes en las fichas técnicas de los artículos, se observó que 511 se encuentran activas en las transacciones realizadas por los usuarios.

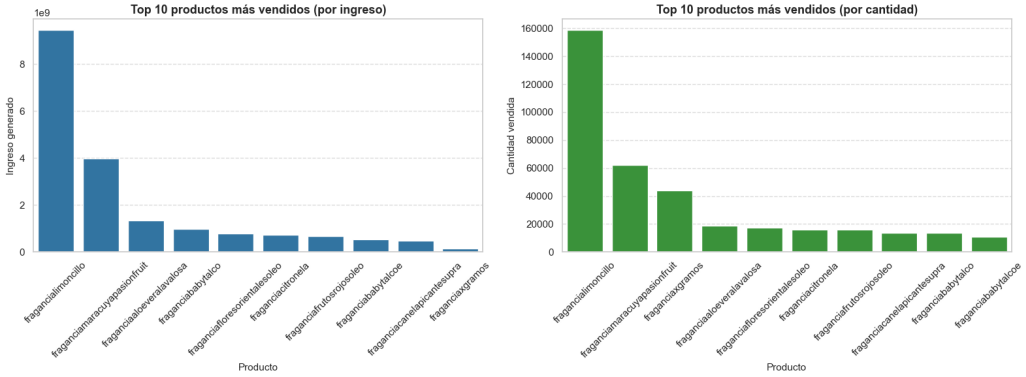

Figura 4.6: Top 10 distribución por producto

Se observa que en ambas gráficas *fragancia limoncillo* es el primer producto, genera la mayor parte del dinero (+8 mil millones).

4.1.5. Análisis geográfico

Se puede visualizar en la siguiente figura el top 10 del valor total y la cantidad de ventas realizadas en cada una de las ciudades y municipios registrados por cada compra realizada por los usuarios.

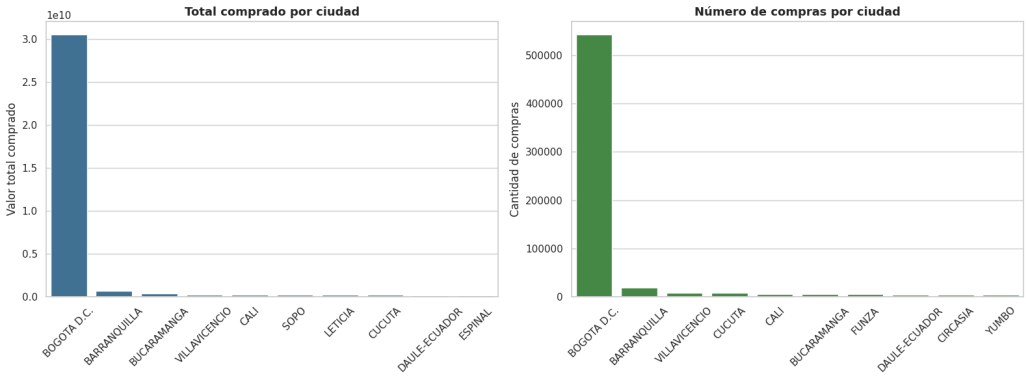

Figura 4.7: Distribución geográfica de las compras

Se observa que la ciudad de Bogotá D.C. domina las ventas con más de 3 mil millones en monto total y más de 500 mil unidades vendidas, siendo la ciudad principal para la empresa.

4.2. Modelo RFM

Se obtuvo la siguiente estadística descriptiva del modelo RFM más las demás variables:

Cuadro 4.3: Estadística descriptiva de las variables del modelo

	Recencia	**Frecuencia**	**Monetario (M)**	**Cantidad**	**Num. Prod.**	**Compra Prom.**	**Ticket Prom. (M)**	**Finde %**
Count	1065	1065	1065	1065	1065	1065	1065	1065
Mean	870.49	19.28	29.21	525.44	32.80	11.82	1.15	0.02
Std	598.73	20.24	59.33	1,005.67	28.29	25.52	1.92	0.10
Min	1	1	0.02	0.67	3	0	0.01	0
25 %	347	6	1.11	22.25	11	0.01	0.14	0
50 %	697	13	5.80	120	24	2.97	0.45	0
75 %	1,435	26	27.70	532.90	47	13.03	1.34	0
Max	2,539	193	656.51	8,997.65	158	428.33	17.74	1

*Valores monetarios expresados en millones (M).

Después de eliminar los usuarios con menos de 3 compras el conjunto de datos queda con 67,182 registros.

Al analizar el cuadro 4.3 y la figura 4.8 se concluye que en general las variables presentan alta asimetría hacia la derecha, con algunos clientes que realizan compras muy altas en valor monetario y cantidad. Es importante realizar diferentes tipos de transformaciones y normalizaciones hasta encontrar la que presente mejores resultados en la disminución de la asimetría y la curtosis para realizar el clustering.

También se observa que los clientes suelen comprar varios tipos de artículos del catálogo, esto permite entender que es viable realizar un PRS, pues el comportamiento indica que se venden artículos diferentes y no siempre los mismos.

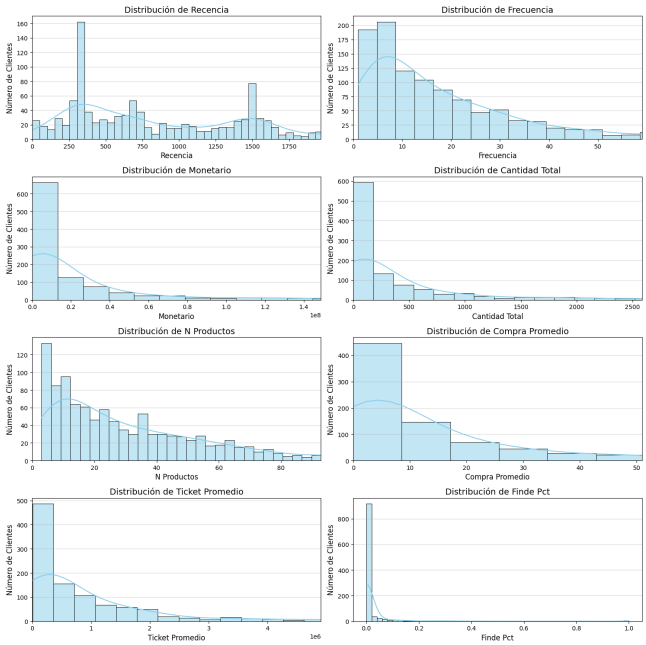

Figura 4.8: Distribución de RFM con más variables

A continuación, se presentan las correlaciones entre las variables utilizando el coeficiente de Spearman, debido a que las variables presentan sesgos y valores atípicos:

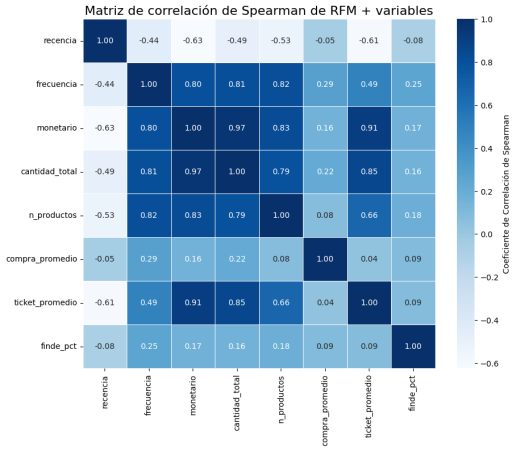

Figura 4.9: Correlaciones de RFM con más variables

La matriz muestra alta colinealidad entre las variables (monetario, ticket_promedio y cantidad_total) y (frecuencia y n_artículos). Por el contrario, las variables compra_promedio y finde_pct muestran correlaciones muy bajas con las demás.

4.3. Modelado

En esta sección se muestran los resultados en la etapa del clustering y del PRS.

4.3.1. Clustering

Después de tener el modelo RFM más las otras variables se realiza el clustering, se decidió iterar con diferentes combinaciones de las variables, transformaciones, métodos de escalamiento y PCA. Con el objetivo de explorar a través de las métricas y los perfiles del clustering, que resultados se obtendrían.
En el siguiente cuadro se muestra un resumen de los mejores resultados obtenidos en el proceso, se itera con los algoritmos y los hiperparámetros mencionados en la metodología:

Cuadro 4.4: Resultados de las iteraciones de clustering

Variables	Transformación	Norm./Estand.	PCA	Algoritmo (Hiperparámetros)	Silueta	CH	DB
Todas	NO	NO	SI	Jerarquico (Euclídea, 3)	0.8544	1375.87	0.2683
Todas	NO	NO	NO	GMM (k=3, tied)	0.8212	2124.81	0.5069
RFM	NO	NO	NO	K-means (k=3)	0.7987	2722.71	0.5152
RFM+n_prod+ticket_prom	NO	StandardScaler	SI	BIRCH (k=3, th=0.5)	0.7946	2455.29	0.4962
RFM+n_prod+ticket_prom	LOG(x+1) + BOX-COX	StandardScaler	SI	K-means (k=6)	0.6347	6084.82	0.4428
recencia+monetario+n_prod+cant_total+finde_pct	LOG + BOX-COX	StandardScaler	SI	BIRCH (k=4, th=0.4)	0.5589	994.65	0.6650
recencia+n_prod+cantidad_total	LOG + BOX-COX	StandardScaler	SI	GMM (k=5, diag)	0.5490	4457.68	0.5039

Transformación de las variables RFM más otras

Después de iterar hasta encontrar las variables que representan los comportamientos de los usuarios, se encontró que las variables RFM, el número de artículos y el ticket promedio de compra son la combinación más adecuada. A continuación, se muestran las transformaciones que se realizaron para tratar los valores atípicos y normalizar las variables:

Cuadro 4.5: Transformación y escala al modelo RFM

Variable	**Asimetría**	**Curtosis**	**Etapa**
Recencia	0.4885	-0.8415	Original
Frecuencia	2.9185	14.6441	
Monetario	4.2157	25.0492	
N_productos	1.4379	2.0816	
Ticket_promedio	3.8331	19.2460	
Frecuencia	0.0048	-0.7962	LOG(1+x) + StandardScaler
Monetario	-0.1799	-0.7528	
N_productos	-0.3993	-0.7367	
Recencia	-0.1193	-0.8991	Box-Cox + StandardScaler
Ticket_promedio	0.0059	-0.6684	

Se halló que usando transformaciones LOG(1+x) y Box-Cox, y estandarizando con $Z - score = \frac{x-\mu}{\sigma}$ se obtuvieron los mejores resultados, ya que las variables originales presentaban alta asimetría positiva y curtosis, especialmente en *monetario* con una curtosis = 25.0492, mostrando que las transformaciones con StandardScaler mejoraron la estructura de los datos para realizar clustering.

PCA

Como se mencionó en la metodología se realiza PCA para reducir la dimensionalidad teniendo en cuenta que las variables RFM más el número de artículos y el ticket promedio de compra cumplen con el supuesto de estar correlacionadas.
Se obtuvo que el PCA con el primer componente (PC1) explica el 76.4 % de la varianza total, mostrando que las demás dimensiones no aportan más del 11 % individual, siendo dominante en una dimensión y con una varianza acumulada suficiente para este trabajo. Se muestra la representación de las variables en PC1.

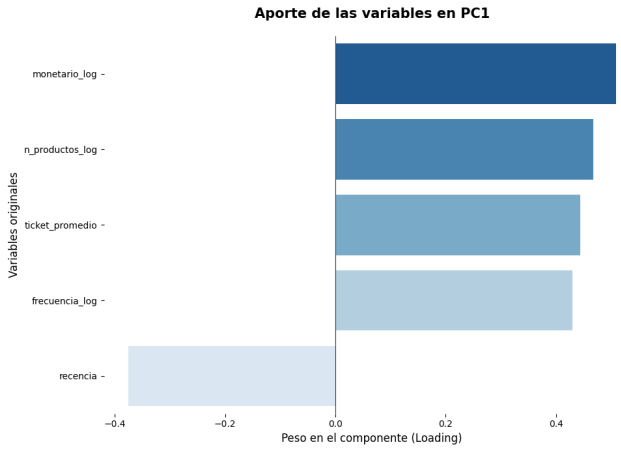

Figura 4.10: Variables en PC1

Se evidencia que el cliente que gasta mucho es casi siempre el mismo que viene seguido, compra variedad de artículos y su última compra es reciente, indicando que un buen cliente

tiene poca recencia. Entre mayor sea la recencia, menor es el gasto del cliente, esta es una variable que se comporta de manera inversa con las demás.

Clustering seleccionado

Como se observa en el cuadro 4.4 el mejor resultado se obtuvo con el algoritmo K-means con k = 6, basándose en los mejores resultados obtenidos en los índices DB, CH y perfiles de los clientes comparado con los demás algoritmos trabajados.

- **Coeficiente de silueta:** 0.6347
- **Índice Davies-Bouldin:** 0.4428
- **Índice Calinski-Harabasz:** 6084.82

Se graficó los clústeres formados en dos dimensiones para identificar la forma visual y la separación entre cada uno:

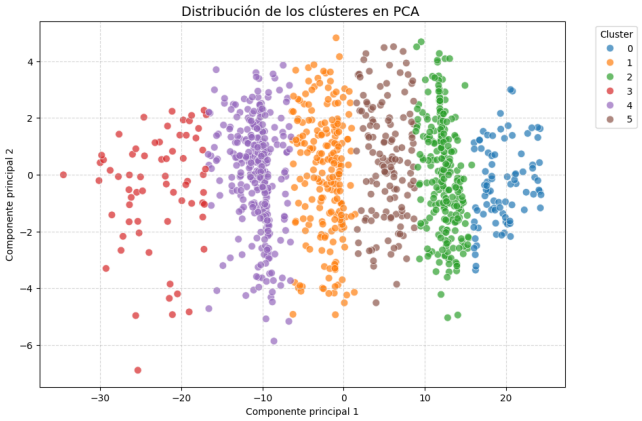

Figura 4.11: Mejor clustering

K-means resultó ser el algoritmo más efectivo porque es muy escalable, agrupando a los usuarios de manera eficiente. Además, la selección de los hiperparámetros permitió crear grupos bien definidos. Por otro lado, el algoritmo jerárquico y DBSCAN no se ajustaron bien a la estructura de los usuarios, con valores en el coeficiente de silueta muy altos, pero con resultados inferiores en los índices y perfiles.
Se realizó un cuadro con el número de usuarios, el promedio de los valores en las variables RFM, cantidad y número de productos para cada clúster.

Cuadro 4.6: Caracterización promedio de los clústeres

Clúster	N. Usuarios	Recencia	Frecuencia	Monetario (M)	Cantidad	Num. Productos
0	86	2038	4	0.39	18	9
1	215	657	15	7.59	115.75	32
2	244	1481.5	7	1.33	50	12
3	67	82	24	55.23	870.15	65
4	318	337	20	23.51	376.19	37.5
5	135	1040	11	4	75.10	24

*Valores monetarios expresados en millones (M).

Análisis de clustering

Con la información del cuadro 4.6 se analiza cada clúster.
La ciudad donde se realizan la mayoría de compras es Bogotá D.C. Sin embargo, se explora la ciudad o municipio que sigue.

- Clúster 0: Presentan la recencia más alta y el valor monetario más bajo, son usuarios inactivos que hace mucho tiempo no realizan transacciones. De acuerdo con los artículos que compraban, se encuentran olores frutales como chicle, fresa o kiwi. Estos son para uso personal, ya que estas fragancias son en presentación pequeña. Cali es la segunda ciudad donde más compran.

- Clúster 1: Compran una gran variedad de artículos del catálogo, con artículos como fragancia x gramos y citronela, indican ser puntos de venta ya que fragancia x gramos corresponde a 25 gramos de todas las referencias del catálogo, con uso para pebeteros y difusores. Leticia es el segundo municipio donde más compran.

- Clúster 2: Presentan la segunda recencia más alta, son usuarios que hace tiempo no interactúan con la empresa. Los artículos que más consumen son fragancia x gramos y canela, indicando ser puntos de venta. Villavicencio es el segundo municipio donde más compran.

- Clúster 3: Presentan la recencia más baja, realizan las compras con el valor monetario y la cantidad más alta de artículos, sus compras son la línea de polímeros, corresponde a plásticos y lavanda talco. Barranquilla es la segunda ciudad donde más compran.

- Clúster 4: Realizan las compras con el valor monetario y la cantidad más alta de artículos después del clúster 3, con la diferencia de que la recencia es más alta. Las compras son limoncillo y maracuyá, representan las empresas que fabrican artículos de aseo como jabones lavaloza. Sopó es el segundo municipio donde más compran.

- Clúster 5: Los usuarios están en riesgo de ser inactivos, lo demuestran a través de sus cifras en recencia y frecuencia. Los artículos más comprados son fragancias blue, suavecito y

aloe. Representan las empresas que fabrican artículos de aseo como jabones de ropa. Daule-Ecuador es la segunda ciudad donde más compran.

4.3.2. PRS

Una vez identificados los perfiles de los clústeres se procede a implementar los PRS.
Como primer paso, se construyeron dos matrices Usuario–Artículo para cada clúster (binaria y conteo de los artículos). Se iteró el cálculo de las similitudes para cada matriz, con el fin de identificar la distancia que mejor represente la similitud entre los usuarios y artículos, para implementar los diferentes filtrados.
Se encontró que los filtrados obtienen mejores resultados con la matriz Usuario–Artículo binaria, usando el cálculo de la similitud del coseno en los PRS de popularidad, IBCF, y la combinación híbrida ponderada de IBCF con contenido.
Al construir las matrices Usuario–Artículo para cada clúster, se obtienen las siguientes dimensiones de los usuarios y artículos: clúster 0 (86, 106), clúster 1 (215, 326), clúster 2 (244, 377), clúster 3 (67, 220), clúster 4 (318, 382) y clúster 5 (135, 321).
Por otro lado, la matriz Artículo–Característica se compone de 511 artículos (filas) y 305 características (columnas).
Para calcular los puntajes de las recomendaciones se sigue la siguiente fórmula:

$$\text{Puntaje}_{\text{Híbrido}}(A_i) = \alpha \cdot Puntaje_{IBCF} + (1 - \alpha) \cdot Puntaje_{Contenido}$$

Para cada clúster se iteró con diferentes valores en el filtrado de ruido y el α, se seleccionó el que obtuviera los mejores resultados en Recall@K y NDCG@K, buscando un equilibrio entre ambas métricas.
En el siguiente cuadro se muestran los mejores resultados obtenidos con sus hiperparámetros:

Cuadro 4.7: Resultados de las iteraciones del PRS

Clúster	Modelo	Hiperparámetro ruido	α	Similitud	Norm./Est.Puntaje	K	n_splits	Recall@K	NDCG@K
0	Popularidad	0.8	N/A	Coseno	Min-Max	20	5	0.5675	0.3506
1	Híbrido	0.25	0.7	Coseno	Z-Score	20	5	0.2352	0.1004
2	Híbrido	0.25	0.85	Coseno	Min-Max	20	5	0.4502	0.2043
3	Basado en artículos	0.25	1	Coseno	Min-Max	20	5	0.5689	0.2267
4	Basado en artículos	0.25	1	Coseno	Min-Max	20	5	0.2802	0.1220
5	Basado en artículos	0.25	1	Coseno	Z-Score	20	5	0.3760	0.1634

La decisión de realizar recomendaciones populares al clúster 0 se tomó a través del perfil que se obtuvo con K-means. Tienen recencia alta y artículos de uso personal, suelen ser usuarios que compran lo que esté de moda o por probar algo nuevo. Realizar estas recomendaciones dió buenos resultados en las dos métricas, demostrando que el perfil y análisis realizado funcionó.
En los procesos de iteración se encuentra que los puntajes de los clústeres 0, 2, 3 y 4 se normalizaron con Min–Max, esto indica que los usuarios tienen un comportamiento de compra estable. Por otro lado, los clústeres 1 y 5 se estandarizaron con Z-Score, ya que los usuarios compran artículos muy diferentes, al por mayor o por unidad. También se encuentra que con $K = 20$

se obtuvieron los mejores resultados, indicando si el producto recomendado se encuentra en la lista de los primeros 20 artículos y en que posición del ranking.
El mejor Recall@20 se obtuvo para el clúster 3 con IBCF (0.5689) y el mejor NDCG@20 para el clúster 0 con popularidad (0.3506).
A continuación, se muestra un ejemplo de las recomendaciones obtenidas en cada clúster para usuarios aleatorios y su respectivo puntaje teniendo en cuenta el tipo de filtrado al que pertenezca el clúster.

Cuadro 4.8: Recomendaciones personalizadas por clúster y usuario

Clúster	**Categoría**	**Fragancia**	**Puntaje**
0 (U.10)	Historial	Durazno	
		Sándalo	
		Ruda invierno	
	Recomendación	Maracuyá passion fruit	**1.00**
		Chicle	**0.86**
		Frutos rojos óleo	**0.85**
1 (U.550)	Historial	Mango néctar	
		Brisa de pino	
		Manzana roja	
		Lavanda	
		Pino multiusos	
	Recomendación	Suavecito	**0.91**
		Vainilla difusor	**0.81**
		Degreemenind	**0.77**
2 (U.367)	Historial	Coco passion	
		Mango néctar	
		Ocean fresh	
		Canela picante	
	Recomendación	Fresita	**0.87**
		Mandarina	**0.81**
		Picadura de tabaco	**0.79**
3 (U.1142)	Historial	Coco passion	
		Brisa de pino	
		Tutti bomba chicle	
		Tutti chicle hidro	
	Recomendación	Ruda invierno óleo	**0.89**
		Suerte rápida	**0.76**
		Pino	**0.71**
4 (U.129)	Historial	Tutti bomba chicle	
		Lavanda	
		Ocean fresh	
	Recomendación	Jazmín	**0.91**
		Jako estándar	**0.82**
		Carro nuevo	**0.81**
5 (U.149)	Historial	Lavanda	
		Pino multiusos	
		Canela picante	
		Piel de durazno	
		Floral extra multi	
	Recomendación	Frutos rojos pn	**0.82**
		Vainilla difusor	**0.59**
		Degreemenind	**0.57**

Se calculó el recall ponderado para evaluar el desempeño del sistema con los 1065 usuarios únicos y se obtuvo un valor de 36.35 %. Esto indica que el sistema es capaz de mostrar en el top 20 uno de cada tres artículos que los usuarios adquieren.
Finalmente, se evaluó la diversidad de los artículos recomendados dentro de cada clúster, a través del conteo de los artículos únicos que se encuentran en las listas Top 20 de cada usuario. Para cada clúster se obtuvieron los siguientes resultados de los artículos únicos recomendados:

Cuadro 4.9: Diversidad de las recomendaciones por clúster

Clúster	**Artículos únicos**
Clúster 0	47
Clúster 1	156
Clúster 2	173
Clúster 3	69
Clúster 4	175
Clúster 5	120

Estos resultados demuestran que los sistemas de recomendación tienen diversidad en los artículos recomendados. En el clúster 0 se evidencia la menor diversidad de artículos, lo que permite confirmar que los usuarios tienen un comportamiento que no varía. Por otro lado, los enfoques híbridos y basados en artículos muestran que el sistema logra ser personalizado y no genera recomendaciones iguales para todos los usuarios, esto incrementa la oportunidad de que descubran nuevos artículos del catálogo.

Capítulo 5

Conclusiones

1. Se realizó el preprocesamiento de 73,089 registros de ventas de los años (2018-2024) y 598 fichas técnicas. Se identificó que la variable ciudad presentaba un 60.14 % de valores nulos, estos fueron imputados por la moda basándose en el vendedor. Las variables numéricas del conjunto de datos del histórico de ventas mostraron alta asimetría positiva y no siguieron una distribución normal, la cual se calculó con la prueba de Kolmogorov-Smirnov. Con esta se realizó la matriz de correlación de Spearman y se observó correlación positiva entre las variables de neto, IVA, cantidad y total. Este análisis sirvió de guía para tratar los valores atípicos descubiertos a través de boxplots e IQR y normalizar las variables una vez construidas las variables del modelo RFM.

2. Se construyó el modelo RFM ampliado, se encontró que las variables RFM con el número de productos y el ticket promedio mostraban alta correlación en la matriz de Spearman. Estas variables presentaban alta asimetría positiva y curtosis, especialmente en monetario. Se halló que usando transformaciones LOG(1+x) y Box-Cox, y estandarizando con Z-score se mejoró la estructura de los datos. Asimismo, teniendo en cuenta la correlación entre las variables se realizó PCA y se encontró que con el primer componente se logró explicar el 76.4 % de la varianza total, mostrando que la recencia se comporta de manera inversa a las demás variables, por lo tanto, un cliente fiel tiene un valor bajo en la recencia.

3. Se identificó que realizar procesos iterativos de los 6 algoritmos de clustering (K-means, GMM, BIRCH, Jerárquico, DBSCAN y Fuzzy C-means) permitió a K-means identificar 6 tipos de usuarios con comportamientos de compra diferentes, demostrando la importancia de crear estrategias específicas a cada segmento y no tratar a todos los usuarios por igual. También se confirma que agregar más variables al modelo RFM guiado de los estudios del cuadro 2.1 del marco teórico facilitó a los algoritmos crear clústeres, evaluar múltiples combinaciones y seleccionar los mejores resultados en las métricas de evaluación, llegando a un coeficiente de silueta de 0.6347, índice de Davies-Bouldin de 0.4428 e índice de Calinski-Harabasz de 6084.82, indicando buena calidad de agrupamiento.

4. Se identificó que analizar el comportamiento de cada clúster es el paso más importante antes de implementar un PRS, ya que permite entender el contexto y descubrir las

recomendaciones más adecuadas. Se iteró implementando PRS basados en popularidad, contenido, IBCF, UBCF e híbrido ponderado dentro de cada clúster. Se construyeron las matrices binarias que requiere cada filtrado y se calculó la similitud de coseno correspondiente. Se encontró que no existe un único algoritmo óptimo para todos los segmentos y se obtuvieron los mejores resultados con popularidad, IBCF e híbrido ponderado de IBCF con contenido, esto confirma el artículo de Beregovskaya y Koroteev (2021), ellos mencionan que estos enfoques son eficientes ante cambios en las preferencias de los usuarios porque las recomendaciones se generan analizando las similitudes entre los artículos y además reduce el costo computacional. Incluir un hiperparámetro de popularidad para reducir el ruido fue fundamental para generar recomendaciones sin sesgos y diversificadas.

5. Se evaluó el rendimiento de los PRS utilizando validación cruzada de 5 pliegues con técnica Leave-One-Out y las métricas Recall@K y NDCG@K con K=20, con el que se obtuvo mejor rendimiento en los clústeres 0 y 3 con NDCG@20 y Recall@20 más altos respectivamente. El recall ponderado general del sistema para los 1065 usuarios fue del 36.35 %, indica que 1 de cada 3 artículos recomendados se encuentran en los primeros 20 resultados de la compra esperada del cliente.

6. En genereal, se logró desarrollar un PRS para datos implícitos validado con el método de validación cruzada con Leave-One-Out y métricas top K con Recall@20 y NDCG@20. Los resultados muestran que K-means logró un mejor desempeño al representar el comportamiento de los clientes con 6 clústeres, lo que permitió construir sistemas de recomendación efectivos. Se usaron los enfoques basados en popularidad, artículos y un método híbrido ponderado que combina contenido y artículos, se alcanzó un recall ponderado del 36.35 %. Estos resultados demuestran que el PRS desarrollado es capaz de generar recomendaciones moderadas para los clientes de la empresa de fragancias, permitiendo mejorar la experiencia de compra e incrementar las ventas.

7. Los modelos de PRS desarrollados lograron generar recomendaciones moderadas para los usuarios a pesar de la ausencia de datos geográficos, limitando captar preferencias contextuales en un entorno real. Se recomienda que en futuros estudios se capture la ubicación durante el registro del cliente para construir PRS basados en los datos geográficos.

8. Teniendo en cuenta la técnica presentada por Isinkaye *et al.* (2015) para el filtrado basado en contenido, como posibles mejoras sería interesante aplicar TF-IDF (Frecuencia de Término-Frecuencia Inversa de Documento), para identificar las características de las fragancias que tienen una relevancia superior, en lugar de la aplicación binaria actual.

Bibliografía

- ANDI - Noticias. (2025). Recuperado de: https://www.andi.com.co/Home/Noticia/17782–en-2024-cada-colombiano-gasto-un-prome
- L'Oréal. (2025, Agosto 6). L'Oréal y NVIDIA impulsan la innovación en belleza con la IA. Recuperado de: https://www.loreal.com/es-es/espana/news/science-and-technology/loreal-y-nvidia-impulsan-innovacion-en-belleza-con-inteligencia-artificial/
- Sephora newsroom. (2026, Marzo 24). Sephora App in ChatGPT Brings a New Personalized Beauty Experience – Sephora Newsroom. Sephora.com. Recuperado de: https://newsroom.sephora.com/sephora-app-in-chatgpt-brings-a-new-personalized-beauty-experience/
- News fragancias. (2022, Noviembre 28). Perfume's Club mejora la experiencia de compra. Revista NewsFragancias. Recuperado de: https://www.newsfragancias.com/profesionales/perfumes-club-nuevos-servicios-compra/
- Iyengar, S. S., & Lepper, M. R. (2000). When choice is demotivating: can one desire too much of a good thing? Journal of Personality and Social Psychology, 79(6), 995–1006. Recuperado de: https://doi.org/10.1037//0022-3514.79.6.995
- Aggarwal, C. C. (2016). Recommender Systems: The Textbook. Springer. doi.org
- Peña, D. (2002). Análisis multivariado de datos. Mc Graw Hill.
- Beregovskaya, I., & Koroteev, M. (2021). Review of clustering-based recommender systems. Recuperado de: http://arxiv.org/abs/2109.12839
- Clustering based algorithms in recommendation system. (2025). GeeksforGeeks. Recuperado de: https://www.geeksforgeeks.org/machine-learning/clustering-based-algorithms-in-recommendation-system/
- De Biasio, A., Navarin, N., & Jannach, D. (2023). Economic Recommender Systems – A Systematic Review. Recuperado de: http://arxiv.org/abs/2308.11998
- Tian Y. *et al.* College library personalized recommendation system based on hybrid recommendation algorithm. Procedia CIRP. - 2019 .– T. 83 .– S. 490-494.
- Bishop C. Pattern Recognition and Machine Learning. New York: Springer-Verlag: 2006. (lnformation Science and Statistics). Recuperado de: //www.springer.com/gp/book/9780387310732
- Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, 16(3), 261–273.Recuperado de: https://doi.org/10.1016/j.eij.2015.06.005
- Eberly College of Science.2.1: Measures of similarity and dissimilarity. (2018). Psu.edu. Recuperado de:

https://online.stat.psu.edu/stat857/node/3/

- Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. Journal of Big Data, 9(1). Recuperado de: https://doi.org/10.1186/s40537-022-00592-5
- How does implicit feedback differ from explicit feedback in recommendations?. Milvus.Io. Recuperado de: https://milvus.io/ai-quick-reference/how-does-implicit-feedback-differ-from-explicit-feedback-in-recommendations
- ¿Qué es el clustering? (2025). IBM. Recuperado de: https://www.ibm.com/es-es/think/topics/clustering
- Rodríguez, D. (2023, junio 9). Método del codo (Elbow method) para seleccionar el número óptimo de clústeres en K-means. Analytics Lane. Recuperado de: http://analyticslane.com/2023/06/09/metodo-del-codo-elbow-method-para-seleccionar-el-numero-optimo-de-clusteres-en-K-means/
- Pacheco, T. D. [Video]. (2024, octubre 23). Machine Learning para Economistas - tutorial 5 - K-fold cross validation. Youtube. Recuperado de: https://www.youtube.com/watch?v=zBHcw22j0n4
- Diagrama de endrograma que representa un: vector de stock (libre de regalías) 2394698767. Shutterstock. Recuperado de: https://www.shutterstock.com/es/image-vector/dendrogram-diagram-representing-tree-hierarchical-cluster-2394698767
- Julio, C. A. C., Cristian, C. H. B., & María, F. L. L. (2025, marzo 26). 3 Clústering Jerárquico. Edu.co. Recuperado de: https://www.icesi.edu.co/editorial/intro-clustering-web/Jerarquico.html
- Kaufman, L., & Rousseeuw, P. J. (1990). Finding groups in data: An introduction to cluster analysis (Vol. 344). John Wiley & Sons.
- Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96) (Vol. 96, pp. 226-231).
- Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. SIGMOD Record, 25(2), 103–114. https://doi.org/10.1145/235968.233324
- Brizuela, K. M. P., & Erazo, A. J. R. (2025). Propuesta de implementación de técnicas de clusterización para la recomendación personalizada de productos para una plataforma de e-commerce en Honduras. Recuperado de: https://repositorio.unitec.edu/server/api/core/bitstreams/2e4416ca-97b7-43bf-acb1-da3532d6a48a/content
- Clustering techniques in recommendations (2025). Meegle.com. Recuperado de: https://www.meegle.com/en_us/topics/recommendation-algorithms/clustering-techniques-in-recommendations
- Ramírez, L. (2024). Algoritmo k-means: ¿Qué es y cómo funciona? Thinking for Innovation. Recuperado de: https://www.iebschool.com/hub/algoritmo-k-means-que-es-y-como-funciona-big-data/
- Raschka, S., Liu, Y. H., & Mirjalili, V. (2022). Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python. Packt Publishing Ltd.
- Noble, J. (2025, febrero 13). ¿Qué es el clustering jerárquico? Ibm.com. Recuperado de: https://www.ibm.com/es-es/think/topics/hierarchical-clustering
- DBSCAN Clustering in ML - Density based clustering. (2019, mayo 6). GeeksforGeeks. Recuperado de:

https://www.geeksforgeeks.org/machine-learning/dbscan-clustering-in-ml-density-based-clustering/

- Bomba, F. [Video]. (2023, mayo 4). SEGMENTACIÓN DBSCAN con KNIME / DBSCAN CLUSTERING ESPAÑOL. Youtube. Recuperado de: https://www.youtube.com/watch?v=I1Zj7MOrz2Y
- GeeksforGeeks. (2018, Agosto 24). Gaussian Mixture Model. GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/gaussian-mixture-model/
- Chaudhry, M., Shafi, I., Mahnoor, M., Ramírez Vargas, D. L., Thompson, E. B., & Ashraf, I. (2023). A systematic literature review on identifying patterns using unsupervised clustering algorithms: A data mining perspective. Symmetry, 15(9), 1679. Recuperado de: https://doi.org/10.3390/sym15091679
- Julio, C. A. C., Cristian, C. H. B., & María, F. L. L. (2025, marzo 26). 2 Generalidades y métricas en el clústering. Icesi Editorial. Recuperado de: https://www.icesi.edu.co/editorial/intro-clustering-web/Metricas.html
- Wolfe, J. H. (1970). Pattern clustering by multivariate mixture analysis. Multivariate Behavioral Research, 5(3), 329–350. https://doi.org/10.1207/s15327906mbr0503_6
- Parsons, M. (2024, julio 14). Transactional segmentation. Experian PLC. Recuperado de: https://www.experian.co.uk/blogs/latest-thinking/guide/rfm-transactional-segmentation/
- Chandana, D. (2024, marzo 18). Exploring Customer Segmentation With RFM Analysis and K-means clustering. Medium.Recuperado de: https://medium.com/thedeephub/exploring-customers-segmentation-with-rfm-analysis-and-k-means-clustering-118f9ffcd9f0
- Sampieri R, Fernandez, C. Baptista, P. Metodología de la investigación científica. Mac Graw Hill. 2003.
- The Python standard library. (s/f). Python Documentation. Recuperado de: https://docs.python.org/3/library/index.html
- Data Science PM. (2024, Diciembre 9). What Is CRISP DM? Data Science Project Management. Recuperado de: https://www.datascience-pm.com/crisp-dm-2/
- Sulca Arango, L. A. (2018). Estadística descriptiva-inferencial Presentación de datos y medidas de dispersión, asimetría y curtosis. Regresión y correlación. Nociones de cálculo combinatorio. Probabilidades. Probabilidad condicional. Probabilidad total. Teorema de Bayes. Variables aleatorias discretas y continuas. Distribuciones de probabilidad. Tipos de distribuciones. Muestreo. Tamaño de una muestra. Aplicaciones con Excel, SPSS y/o Minitab. Intervalo de confianza. Prueba de hipótesis de Z y T. Prueba de hipótesis de X2 y F. Recuperado de: https://repositorio.une.edu.pe/entities/publication/06cf5779-1195-4dc5-9c60-df95aca92ccd
- Hui-Hsin Huang. (2025). Customer segment model with the purchase recency, frequency and monetary amount. Marketing (0354-3471), 56(2), 113–122. Recuperado de: https://doi.org/10.5937/mkng2502113H
- D. Yan, C. -Y. Luo, Y. Li, B. Zhu, M. -L. Yan and S. -L. Yao, Çharging Behavior Analysis Based on BIRCH Clustering,"2022 12th International Conference on Power and Energy Systems (ICPES), Guangzhou, China, 2022, pp. 450-454, doi: 10.1109/ICPES56491.2022.10072610.
- Anitha, P., & Patil, M. M. (2022). RFM model for customer purchase behavior using K-Means algorithm.Journal of King Saud University - Computer and Information Sciences,34(5), 1785–1792. https://doi.org/10.1016/j.jksuci.2019.12.011
- Darraz, N., Karabila, I., El-Ansari, A., Alami, N., & El Mallahi, M. (2025). Integrated sentiment analysis

with BERT for enhanced hybrid recommendation systems. Expert Systems With Applications, 261(125533), 125533. https://doi.org/10.1016/j.eswa.2024.125533

- Chen, D., Sain, S. & Guo, K. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining.J Database Mark Cust Strategy Manag 19, 197–208 (2012). https://doi.org/10.1057/dbm.2012.17

- Rungruang, C., Riyapan, P., Intarasit, A., Chuarkham, K., & Muangprathub, J. (2024). RFM model customer segmentation based on hierarchical approach using FCA.Expert Systems With Applications,237(121449), 121449. https://doi.org/10.1016/j.eswa.2023.121449

- Gomes, M. A., & Meisen, T. (2023). A review on customer segmentation methods for personalized customer targeting in e-commerce use cases. Information Systems and E-Business Management, 21(21), 527–570. https://doi.org/10.1007/s10257-023-00640-4

- Santoyo, A. M., Montes Dorantes, P. N., Pérez Estrada, E. G., Carmona Frausto, J. C., Cervantes Alvarez, S., & Computacionales, S. (2023). Una mejora del algoritmo Fuzzy c-Means utilizando la función de distribución gaussiana. Edu.mx. Recuperado de: https://www.itvictoria.edu.mx/oferta/posgrado2025/msistemas/Otros%20productos/2023%20Revista%20CIM%20Me

- Dedi, Dzulhaq, M. I., Sari, K. W., Ramdhan, S., Tullah, R., & Sutarman. (2019). Customer segmentation based on RFM value using K-means algorithm.2019 Fourth International Conference on Informatics and Computing (ICIC), 1–7.

- Chavhan, R., Dutta, P., Samant, N., & Kar, S. (2025). Data-driven strategic customer segmentation considering cart abandonment behavior: Insights from e-grocery delivery platforms.Information Sciences,718(122327), 122327. Recuperado de: https://doi.org/10.1016/j.ins.2025.122327

- Christy, A. J., Umamakeswari, A., Priyatharsini, L., & Neyaa, A. (2021). RFM ranking – An effective approach to customer segmentation.Journal of King Saud University - Computer and Information Sciences,33(10), 1251–1257. Recuperado de: https://doi.org/10.1016/j.jksuci.2018.09.004

- Chen, R., Fan, J., & Wu, M. (2024). ABNS: Association-based negative sampling for collaborative filtering.Expert Systems With Applications,250(123868), 123868. Recuperado de: https://doi.org/10.1016/j.eswa.2024.123868

- Stylianou, T., & Pantelidou, A. (2025). A machine learning approach to consumer behavior in supermarket analytics.Decision Analytics Journal,16(100600), 100600. Recuperado de: https://doi.org/10.1016/j.dajour.2025.100600

- Hwangbo, H., Kim, Y. S., & Cha, K. J. (2018). Recommendation system development for fashion retail e-commerce.Electronic Commerce Research and Applications,28, 94–101. Recuperado de: https://doi.org/10.1016/j.elerap.2018.01.012

- Shinde, S. K., & Kulkarni, U. (2012). Hybrid personalized recommender system using centering-bunching based clustering algorithm.Expert Systems With Applications,39(1), 1381–1387. Recuperado de: https://doi.org/10.1016/j.eswa.2011.08.020

- Sciencedirect.com. Recuperado el 12 de febrero de 2026, de https://www.sciencedirect.com/ai

- scikit-learn. (2010). 2.3. Clustering — scikit-learn 0.20.3 documentation. Scikit-Learn.org. Recuperado de: https://scikit-learn.org/stable/modules/clustering.html

- Han, J., Kamber, M., & Pei, J. (2012). Cluster Analysis. Data Mining, 443–495. Recuperado de: https://doi.org/10.1016/b978-0-12-381479-1.00010-1

- Klimashevskaia, A., Jannach, D., Elahi, M. et al. A survey on popularity bias in recommender systems. User Model User-Adap Inter 34, 1777–1834 (2024). Recuperado de: https://doi.org/10.1007/s11257-024-09406-0

- GeeksforGeeks. (2022, Febrero 17). How to Calculate Skewness and Kurtosis in Python? GeeksforGeeks. Recuperado de: https://www.geeksforgeeks.org/data-science/how-to-calculate-skewness-and-kurtosis-in-python/

- Zan, S., Zhang, Y., Meng, X., Lv, P., & Du, Y. (2021). UDA: A user-difference attention for group recommendation. Information Sciences, 571, 401–417. Recuperado de: https://doi.org/10.1016/j.ins.2021.04.084

- Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian Personalized Ranking from Implicit Feedback. Arxiv.org. Recuperado de: https://arxiv.org/abs/1205.2618

- Chen, H., Qian, F., Chen, J., Zhao, S., & Zhang, Y. (2021). Attribute-based Neural Collaborative Filtering. Expert Systems with Applications, 185, 115539. Recuperado de: https://doi.org/10.1016/j.eswa.2021.115539