UNIVERSIDAD
NACIONAL
DE COLOMBIA

TRABAJO DE GRADO

ANÁLISIS DE CLÚSTER AUTOMATIZADO

Marisol Correa Henao

mcorreah@unal.edu.co

Juan David Velásquez

jdvelasq@unal.edu.co

Director

Universidad Nacional de Colombia - Sede Medellín

Facultad de Minas

Medellín

2021

Resumen

En este documento se desarrolla el proceso de software de análisis de clúster automático, aunque en la actualidad, existen varias librerías que permiten realizar análisis de clúster, se busca automatizar el proceso y lograr diferentes opciones centralizadas en un mismo paquete; facilitando el análisis y la parametrización de los modelos.

Para su elaboración, se utilizaron las librerías ya existentes en Python, tomando como base lo que se tiene en diferentes herramientas y software estadístico o de análisis de datos, de manera que se puedan usar tanto por una persona con conocimientos básicos como por una persona con conocimientos profundos que quiera parametrizar sus análisis.

Los resultados de este trabajo muestran que es posible facilitar los procesos de agrupamiento y su respectivo análisis de datos a través de los algoritmos actuales, guiando al usuario de manera simple, gráfica, intuitiva en todo el proceso, llevando a concluir que los resultados del análisis de clúster se ve sujeto a la subjetividad o a los conocimientos del usuario sin embargo esta subjetividad es posible reducirla a través de estrategias, técnicas, análisis y el buen uso de las herramientas existentes.

Palabras Clave:
Análisis, Clúster, Software, Python, Librería, aprendizaje de máquinas automático.

Abstract

In this document the automatic clúster analysis software process is developed, although at present, there are several libraries that allow clúster analysis to be carried out. The aim is to automate the process and achieve different centralized options in the same package; facilitating the analysis and parameterization of the models.

For its preparation, existing libraries in python were used, taking as a basis what is available in statistical tools and software or data analysis, so that they can be used both by a person with basic knowledge and by a person with knowledge. that you want to parameterize your analysis.

The results of this process show that it is possible to facilitate the grouping results and their respective data analysis through current algorithms, guiding the user in a simple, graphical, intuitive way throughout the process, leading to the conclusion that the results of the analysis Clustering is subject to subjectivity or user knowledge, however this subjectivity can be reduced through strategies, techniques, analysis and the proper use of existing tools.

Key Words
Analysis, Cluster, Software, Python, Library, Automatic Machine Learning

Tabla de contenido

Introducción

Las sociedades primitivas se organizaron para lograr trascender su ámbito local, lo que los llevó a enfrentar la necesidad de tomar decisiones que exigían un conocimiento numérico de los recursos disponibles. Esto dio lugar al desarrollo de las primeras técnicas estadísticas basadas en el recuento y análisis de datos. A partir de allí, la humanidad ha mejorado las técnicas e implementado la estadística para resolver problemas en casi todos los ámbitos de estudio como las ciencias, la sociedad y la economía (Fernández, Sánchez, Córdoba, & Largo, 2002).

Una de las herramientas más utilizadas en diferentes disciplinas es el análisis clúster, el cual comprende diferentes técnicas, dado un conjunto de variables, se obtienen subconjuntos, teniendo en cuenta criterios de homogeneidad interna, pero guardando una diferencia significativa entre los conjuntos.

En la actualidad, existen diversas herramientas de software para el análisis clúster o de conglomerados, sin embargo, presentan diferentes desventajas que hacen que no sean de fácil manejo para usuarios inexpertos con la programación o el conocimiento estadístico, lo que ocasiona que muchas veces no se logren tener los resultados necesarios para la toma de decisiones.

Por lo anterior, el objetivo de este trabajo es desarrollar un prototipo de librería que permita análisis automático de clústers mediante las metodologías y funcionalidades descritas en el documento y que facilite las decisiones de modelado por parte del usuario.

Dado esto, se pretende realizar una contribución transversal a diferentes disciplinas que utilizan el análisis clúster para la toma de decisiones, creando una herramienta con la que puedan lograr sus objetivos, a pesar de no tener un conocimiento avanzado en programación o en estadística, realizando sus parametrizaciones de manera automatizada.

En este documento se encontrará en primer lugar, un esquema general con los metadatos del código. En segundo lugar, la motivación y la importancia en la que se enmarca este trabajo, presentando una breve descripción de las diferentes técnicas y herramientas de análisis de clúster encontrados en la literatura, y planteando un comparativo de algunas de las principales librerías que existen en las herramientas descritas con sus características y oportunidades de mejora. En tercer lugar, se describe el software desarrollado para análisis de clúster y sus ventajas respecto a las herramientas disponibles mostrando así el aporte realizado, el cual se basa en el desarrollo de un software intuitivo, útil y sencillo para cualquier tipo de usuario. En cuarto lugar, se encuentran diferentes casos de usos comparativos con diferentes herramientas. Por último, se presentan las conclusiones.

1. Metadatos Requeridos

Tabla 1. Metadatos del Código

Nr	Metadatos del código	Descripción
C1	Versión actual del código	v 0.1.0
C2	Repositorio del proyecto	https://github.com/marcorhe/TDGMarisolCorreaHenao
C3	Licencia	MIT
C4	Sistema de control de versiones	git
C5	Lenguaje programación de software, herramientas y servicios utilizados	Python, pandas, numpy, scikit-learn, yellowbrick, seaborn, warning, sys, matplotlib, time, jupiter, ipywidgets.
C6	Requisitos de compilación, entornos operativos y dependencias	Python 3.7.1 o mayor
C7	Documentación y manuales de uso	https://github.com/marcorhe/TDGMarisolCorreaHenao/tree/main/docs
C8	Correo electrónico de soporte para preguntas	mcorreah@unal.edu.co

Fuente: Elaboración propia

2. Motivación e importancia

2.1. Definición problema clustering

Dentro de la evolución de las necesidades para el análisis de datos, surgió a principios del siglo XXI el término ciencia de datos atribuido a Cleveland (2001), quién lo define como una continuación de algunos campos de análisis de datos como la estadística, pone la nueva disciplina propuesta en el contexto de la informática y el trabajo contemporáneo en la minería de datos.

Se define la ciencia de datos como un campo que combina las ciencias de la computación, la estadística, las matemáticas, y la experiencia en las ciencias del comportamiento para la toma de decisiones; sin embargo, se acepta que la inclusión de la investigación de operaciones en la ciencia de datos da origen a la analítica (Hazen et al.,2014). Otro elemento agregado posteriormente es la relación de la ciencia de datos con la minería de datos (Provost, Fawcett, 2013) y el aprendizaje automático (Chojnacki et al., 2017).

Desde la estadística, se presentan dos ramas principales para el análisis de datos, la estadística descriptiva relacionada con la descripción, y la estadística inferencial relacionada con la

extracción de conclusiones a partir de los datos. Existen diferentes métodos estadísticos para analizar un conjunto de datos; como lo son el análisis univariante, el cual es definido por Cuadras como la descripción de una sola variable mediante características muestrales, o la realización de inferencias sobre la información de una muestra obtenida bajo una o varias condiciones experimentales; y el análisis multivariante que cumple también con los objetivos descriptivos e inductivos, pero trabajando simultáneamente con varias variables en lugar de una sola, convirtiéndola en una metodología estadística más compleja, pero al mismo tiempo más potente (Van der Aalst, 2016).

Existen diversos objetivos que motivan la realización de análisis de datos multivariantes, entre ellos, Peña (2002) señala los siguientes:

- Reducir el número de variables a utilizar, minimizando la pérdida de información; se busca utilizar únicamente la cantidad necesaria de variables para explicar el fenómeno analizado.
- Agrupar los datos disponibles según similitudes entre las propiedades de estos, con el fin de analizar las características de cada grupo.
- Clasificar nuevas observaciones en grupos que ya han sido determinados previamente.

En línea con lo anterior, este trabajo se enfoca en la agrupación, específicamente en el análisis clúster, también conocido como análisis de conglomerados, se engloba dentro de los métodos conocidos como interdependientes. Se trata de una técnica creada para el descubrimiento de grupos de observaciones que tienen características similares. Persigue reunir las observaciones en grupos lo más homogéneos posible, de manera que los elementos integrantes de los conglomerados sean muy parecidos. Al mismo tiempo, se busca la máxima heterogeneidad entre grupos (Uriel, & Aldás, 2005).
En la literatura, también es común encontrar el término clústering, que hace referencia a diferentes técnicas utilizadas para encontrar patrones o grupos dentro de un conjunto de observaciones. Estos grupos son llamados clústers y dada la utilidad encontrada en los agrupamientos para ser aplicada en diversas disciplinas, se ha desarrollado gran variedad de métodos y algoritmos para su uso, de los cuáles pueden diferenciarse tres grupos principales:

- **Métodos Jerárquicos:** Este tipo de algoritmos no requieren que el usuario especifique de antemano el número de **clústers**. (agglomerative clustering, divisive clustering).
- **Métodos No Jerárquicos:** Este tipo de algoritmos requieren que el usuario especifique de antemano el número de clústers que se van a crear (K-means, K-medoids, CLARA).
- **Otros métodos:** Métodos que combinan o modifican los anteriores (hierarchical K-means, fuzzy clustering, model based clustering y density based clustering).

Aldenderfer y Blashfield (1984) dividen el análisis clúster en etapas, las cuales se pueden desglosar de la siguiente manera:

- Selección de la muestra que se dividirá en grupos durante el análisis.

- Definición de las variables que van a ser incluidas en la muestra.
- Definición de la métrica de distancia a utilizar.
- Selección de la metodología para agrupar los datos.
- Aplicación del algoritmo de agrupamiento de similitudes o distancias entre elementos.
- Análisis de resultados.

2.2. Medidas de distancia

Para poder determinar agrupaciones de datos es necesario definir una medida de similitud o diferencia entre observaciones (Rodrigo, 2020), se definen algunas de las más utilizadas:

2.2.1. Euclídea:

Es la distancia ordinaria entre dos puntos; cabe mencionar que es la más usada y sencilla. Se calcula a través del teorema de Pitágoras, bajo la siguiente ecuación:

Sean p y q dos puntos del plano cartesiano de coordenadas (x, y), la distancia D Euclídea es:

$$D(p,q) = \sqrt{(p_x - q_x)^2 + \left(p_y - q_y\right)^2}$$

2.2.2. Manhattan:

Este método es una alternativa a la distancia euclídea, el cual permite menos afectación por datos atípicos, ya que no se eleva al cuadrado las diferencias, sino que se mide el valor absoluto de la siguiente manera:

Sean p y q dos puntos del plano cartesiano de coordenadas (x,y), la distancia D manhattan es:

$$D(p,q) = |p_i - q_i|$$

2.2.3. Mahalanobis:

Este método se utiliza para darle más sentido estadístico a la distancia euclídea, teniendo en cuenta que la correlación entre variables, esta medida es invariante respecto a cambios de escala.
De manera formal, la distancia de Mahalanobis entre dos variables aleatorias con la misma distribución de probabilidad $\vec{x}$ y $\vec{y}$ con matriz de covarianza Σ se define como:

$$d_m(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T \Sigma^{-1} (\vec{x} - \vec{y})}.$$

Si la matriz de covarianza es la matriz de identidad, la distancia de Mahalanobis se reduce a la distancia euclidiana. Si la matriz de covarianza es diagonal, entonces la medida de distancia resultante se llama distancia euclidiana estandarizada:

$$d(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{s_i^2}},$$

Dónde S_i es el Desviación Estándar del x_i y y_i sobre el conjunto de muestra.

2.2.4. Simple matching coefficient

En el caso de las medidas categóricas, uno de los métodos más utilizados para medir distancias es el simple matching coefficient. Se utiliza cuando se pretende determinar la similitud entre observaciones de tipo binario, ya que permite calcular la distancia sin utilizar operaciones aritméticas. Se calcula teniendo en cuenta la siguiente ecuación:

Dadas dos observaciones A y B, cada uno con n atributos binarios, el simple matching coefficient (SMC) define la similitud entre ellos como:

$$SMC = \frac{\text{número coincidencias}}{\text{número total de atributos}} = \frac{M_{00} + M11}{M_{00} + M_{01} + M_{10} + M_{11}},$$

donde M_{00} y M_{01} son el número de variables para las que ambas observaciones tienen el mismo valor (ambas 0 o ambas 1), y M_{01} y M_{10} el número de variables que no coinciden. El valor de distancia simple matching distance (SMD) se corresponde con 1−SMC.

2.2.5. Índice Jaccard

Este método es similar al simple matching coefficient (SMC), la diferencia radica en que, el SMC, tiene el término M_{00} en el numerador y denominador, mientras que el índice de Jaccard no. Esto significa que SMC considera como coincidencias tanto si el atributo está presente en ambas partes, como si el atributo no está en ninguno de estos; mientras que Jaccard solo cuenta como coincidencias cuando el atributo está presente en ambos.

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}},$$

o en términos matemáticos de conjuntos:

$$J(A,B) = \frac{A \cap B}{A \cup B}$$

La distancia de Jaccard ($1 - J$ supera a la simple matching distance en aquellas situaciones en las que la coincidencia de ausencia no aporta información. Por ejemplo, suponga que se desea cuantificar la similitud entre dos personas que compran en una tienda, basados en los productos que compraron. Se espera que cada persona adquiere unos pocos productos de los muchos disponibles, por lo que el número de productos no comprados por ninguno de los dos (M_{00}) será

muy alto. Dado lo anterior, como la distancia de Jaccard ignora las coincidencias de tipo M_{00} El grado de similitud dependerá únicamente de las coincidencias entre los productos comprados.

2.3. Principales algoritmos

Existen diferentes algoritmos para definir las agrupaciones, a continuación, se presentan algunos de los más usados (Rodrigo, 2020; Scikit-learn, S.F.).

2.3.1. K-means:

Este algoritmo agrupa las observaciones en un número predefinido de K clústers de tal forma que la suma de las varianzas internas de los clústers sea lo menor posible (MacQueen, J. B., 1967).

2.3.2. K-medoids:

Es un método de clustering que al igual que K-means, agrupa las observaciones en K clústers, donde K es un valor preestablecido por el usuario. En este método cada clúster está representado por una observación presente, a la cual se le llama *medioide*, la cual es el elemento más central del clúster, considerándose el más representativo, cuya distancia promedio entre él y todos los demás elementos del mismo clúster es lo mejor posible.

La diferencia entre K-medoids y K-means, es que el primero se centra en la observación del elemento medioide, mientras que en el segundo cada clúster está representado por su centroide, el cual se corresponde al promedio de todas las observaciones del clúster y no con ninguna en particular. Sin embargo, el hecho de utilizar medoids en lugar de centroides hace de K-medoids un método más robusto que K-means, viéndose menos afectado por outliers o ruido.

2.3.3. K-prototypes:

Es un algoritmo de agrupación restringido, el cual permite agrupar grandes grupos de datos mixtos, numéricos y categóricos. Como el clustering se realiza con base en la comparación de los objetos de K prototipos, en vez de K-medias (means), se le llama k-prototypes.

El aporte que este método hace al método K-means, es permitir el análisis de datos categóricos, preservando su eficiencia.

2.3.4. Aglomerativo:

Es un método jerárquico, es decir que no requiere pre-especificar el número de clústers, donde el agrupamiento inicia con todas las observaciones separadas, cada una formando un clúster individual. A medida que la estructura crece hasta converger en uno solo, los clústers se van combinando.

2.3.5. Density based clustering, Dbscan

Este método fue presentado por Ester et al (1996) como una forma de identificar clústers, siguiendo el modo intuitivo en el que lo hace el cerebro humano, identificando regiones con alta densidad de observaciones separadas por regiones de baja densidad.

La diferencia de este método con los métodos de partición como k-means, jerárquicos y k-medoids, es que estos son buenos encontrando agrupaciones con formas esféricas o convexas que no contengan un exceso de outliers o ruido, pero fallan al tratar de identificar otro tipo de formas más arbitrarias. Dado esto, el Dbscan evita esto, siguiendo la idea que para que una observación haga parte de un clúster, tiene que haber un mínimo de observaciones vecinas dentro de un radio determinado de proximidad, adicionalmente, considera que los clústers están separados por regiones vacías o con pocas observaciones.

2.3.6. Fuzzy

Este método es uno de los más utilizados en la lògica difusa, es una técnica de minería de datos para el clustering que se basa en el algoritmo clásico C-Means, cuyo objetivo es optimizar el agrupamiento difuso, en el cual, los puntos de datos pueden pertenecer a más de un grupo, y se asocia a cada uno de los puntos un indicador correspondiente al grado en que los puntos de datos pertenecen a los diferentes grupos. En términos generales, el método consiste en minimizar iterativamente la función hasta obtener una partición difusa óptima.

2.4. Métricas para determinar el número de grupos

Uno de los problemas más comunes al aplicar alguno de los métodos de clustering como k-means y k-mediods, es el hecho de establecer el número de clústers. Esto es importante, ya que si se definen pocos clústers pueden causar problemas de heterogeneidad dentro de los mismos; y si, por el contrario, se definen muchos clusters, pueden causar que datos muy similares sean divididos en más grupos de lo necesario.

Dado lo anterior, aunque no existe un único criterio ampliamente válido para determinar un número óptimo de clústers, en la literatura se pueden identificar diferentes métodos que pueden resolver este problema; como lo son el método del codo, el criterio de Calinski-Harabasz y el método de la silueta.

2.4.1. Método del codo

Ajusta el modelo con un rango de valores para K. Si el gráfico de líneas se asemeja a un brazo, entonces el "codo" (el punto de inflexión en la curva) es una buena indicación de que el modelo subyacente se ajusta mejor en ese momento.

Para encontrar un “codo” visual, que es el número óptimo de grupos, debe graficar la distancia interna promedio por grupo de la suma de cuadrados frente al número de grupos, siendo matemáticamente:

$$W_k = \sum_{r=1}^{k} \frac{1}{n_r} D_r$$

Donde $k$es el número de clústers, $n_r$es el número de puntos en el grupo r y $D_r$es la suma de las distancias entre todos los puntos del grupo:

$$D_r = \sum_{i=1}^{n_r-1} \sum_{j=i}^{n_r} ||d_i - d_j||_2$$

2.4.2. Criterio de Calinski-Harabasz

Es un índice de validación interna para probar la convergencia dentro de los clústers, y la divergencia entre los mismos en cada nivel de aglomeración del proceso jerárquico (Introini y Lena, 2011). La forma de estimar el índice Calinski-Harabasz, se puede observar en la siguiente ecuación, dada para n número de observaciones y k número de clústers:

$$CH = \frac{Dispersión\ B/(k-1)}{Dispersión\ W/(k-1)}$$

Donde B es la matriz de dispersión entre los clústers y W es la matriz de dispersión dentro del clúster (Maulik y Bandyopadhyay, 2002). Estas dos últimas se pueden expresar como:

$$Dispersión\ B = \sum_{k=1}^{K} n_k \|z_k - z\|^2 \qquad Dispersión\ W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \|x_i - z_k\|^2$$

Donde n_k es el número de observaciones en el clúster k, z_k es la media o centroide del clúster k y z es la media de toda la base de datos (Maulik & Bandyopadhyay, 2002). La partición que maximice el índice de Calinski-Harabasz será la que cree clústers más homogéneos.

2.4.3. Método de la silueta

Este análisis mide qué tan bien agrupada está una observación y estima la distancia media entre clústers. El gráfico de silueta muestra una medida de qué tan cerca está cada punto en un clúster de los puntos en los clústeres vecinos, por lo que proporciona una manera de evaluar parámetros como el número de clústeres de forma visual.

Para cada observación i, el ancho de la silueta $S_i$se calcula así:

- En primer lugar, para cada observación i, calcula la disimilitud media $a_i$entre iy todos los demás puntos del clúster al que pertenece.
- En segundo lugar, para todos los demás clústers Ca los cuales no pertenece, calcule la disimilitud media $d(i, C)$de i para todas las observaciones de C. El más pequeño de estos $d(i, C)$se define como $b_i = min_c d(i, C)$. El valor b_i puede verse como la diferencia entre iy su grupo "vecino", es decir, el más cercano al que no pertenece.

- Finalmente, el ancho de la silueta de la observación. $i$está definido por la fórmula S_i=(b_i-a_i)/ $max(a_i,b_i)$.

El ancho de la silueta se puede interpretar así:

- Las observaciones con un $S_i$grande (casi 1) están muy bien agrupadas.
- Un S_i pequeño (alrededor de 0) significa que la observación está entre dos grupos.
- Las observaciones con un $S_i$negativo probablemente están ubicadas en un grupo equivocado.

2.5. Librerías disponibles

En el mercado se encuentran diferentes herramientas como hadoop, especialmente spark y librerías como scikit-learn han hecho una apuesta para trabajar con grandes volúmenes de datos, permitiendo que más áreas de la ciencia se incorporen a realizar sus análisis y al aprovechamiento de datos y logrando una optimización de los modelos que se implementan.

Adicionalmente, se cuenta con herramientas diseñadas específicamente para permitir una interacción gráfica con el usuario, como es el caso de R con paquetes como rattle o herramientas como Orange data mining, rapidminer, SPSS, SAS, entre otras; las cuales, presentan un asistente de clustering que pide una parametrización del tipo de clúster a implementar como cantidad de grupos, tratamiento a realizar en los datos y la salida esperada; siendo un entorno gráfico más amigable para el usuario.

La estadística ha estado dedicada tradicionalmente al análisis de pequeños volúmenes de datos, ya que la disponibilidad de grandes volúmenes de datos se ha dado únicamente en la última década. La prospección de datos representa un reto para la estadística, debido a la necesidad de crear instrumentos de trabajo que den respuestas de fácil comprensión a usuarios no siempre expertos en su funcionamiento. De hecho, buena parte de los métodos de prospección de datos proceden de la estadística; por ejemplo, los métodos de clasificación y agregación de datos, los modelos de predicción, las redes bayesianas y los métodos heurísticos (métodos de investigación que inventan nuevos procesos, nuevas formas de organización, a partir de los que se están llevando a cabo con el objetivo de mejorarlos).

Las ciencias y la economía presentan una dependencia notoria de la estadística. Para poder analizar cualquier realidad científica, social o económica de interés es imprescindible la utilización de diferentes métodos estadísticos que permitan la observación del fenómeno y la recolección de datos. Dado esto, se busca lograr una mayor compresión de la realidad analizada que permita la toma de decisiones.

Lo anterior, aunado con la nueva tendencia de automatización del trabajo está empezando a permear en las empresas gracias, en gran medida, a las tecnologías y técnicas de Machine

Learning, pero también intervienen otros factores como la incorporación de avances en interacción, información e integración (SAS, 2018).

En los diferentes lenguajes de programación y software estadísticos se encuentran paquetes para análisis de clúster o agrupamiento que permiten al usuario realizar sus agrupaciones, cada uno presenta ventajas y desventajas. Dadas esto, con este trabajo, se plantea presentar una solución que permita mitigar esas situaciones negativas. Para ilustrarlo con más detalle, se construye la siguiente tabla, donde se hace un comparativo entre las diferentes herramientas existentes con base en la experiencia del uso de estos programas:

Tabla 2. Comparativo de librerías.

Librería	**Algoritmos implementados**	**Parámetros**	**Métrica de distancia**	**Observaciones**
Scikit-learn (Python)	K-Means	Número de clústers	Euclidean, Manhattan, Chebyshev, Minkowski, Mahalanobis	Escalabilidad muy grande n_samples, mediano n_clusterscon código MiniBatch
	Affinity propagation	Amortiguación, preferencia de muestra	Haversine	No escalable con n_samples
	Mean-shift	Ancho de banda	Euclidean, Manhattan, Chebyshev, Minkowski, Mahalanobis	No escalable con n_samples
	Spectral clustering	Número de clústers	Haversine	Escalabilidad mediano n_samples, pequeñon_clusters
	Ward	Número de conglomerados o umbral de distancia	Euclidean, Manhattan, Chebyshev, Minkowski, Mahalanobis	Escalabilidad grande n_samplesyn_clusters
	Hierarchical	Número de conglomerados o umbral de distancia, tipo de enlace, distancia	pairwise	Escalabilidad grande n_samplesyn_clusters

	DBSCAN	Distancia al vecino más cercano	Euclidean, Manhattan, Chebyshev, Minkowski, Mahalanobis	Escalabilidad muy grande n_samples, medianon_clusters
	OPTICS	Cantidad mínima observaciones por clúster	Euclidean, Manhattan, Chebyshev, Minkowski, Mahalanobis	Escalabilidad muy grande n_samples, granden_clusters
	Gaussian mixtures	Muchos	Mahalanobis	No escalable
	BIRCH	Factor de ramificación, umbral, agrupador global opcional.	Euclidiana	Escalabilidad grande n_clustersyn_samples
MLlib (Spark)	K-Means	Número de grupos, número máximo de iteraciones, el umbral de distancia, conjunto opcional de centros de clúster que se utilizan para la inicialización	Euclidean	Escalabilidad muy grande n_samples, mediano n_clusterscon código MiniBatch
	Gaussian mixtures	Número de clusters, cambio máximo en la probabilidad logarítmica en el que consideramos la convergencia lograda, número máximo de	Euclidean, Mahalanobis	No escalable

		iteraciones que se pueden realizar sin alcanzar la convergencia, punto de partida opcional desde el cual iniciar el algoritmo EM.		
	Power iteration clustering (PIC)	Número de clusters, número máximo de iteraciones, modelo de inicialización	Euclidean	Escalable
	Latent Dirichlet allocation (LDA)	Número de clusters, Optimizador para usar para aprender el modelo LDA, Parámetro de Dirichlet para distribuciones anteriores de documentos sobre temas, número de iteraciones.	Euclidean	Escalable
	Bisecting	Número de clúster, número máximo de iteraciones, número mínimo de puntos, semilla aleatoria	Euclidean	Escalable

	Streaming	Número de clúster	Euclidean	Escalable
Mahout	K-Means	Número de clúster, número máximo de iteraciones,métrica distancia	Chevychev, coseno	Escalable
	Canopy	Distancia mínima y máxima de mapeo, métrica de distancia	Chevychev, coseno	No escalable con vectores de otro tipo
	Latent Dirichlet allocation (LDA)	Número de clusters, delta convergencia, número de iteraciones.	Chevychev, coseno	Escalable
	Fuzzy	Número de clúster, métrica de distancia, parámetro fuzzy, número máximo de iteraciones, delta de convergencia	Chevychev, coseno	Escalable
	Streaming	Número de clúster, métrica de distancia, número máximo de iteraciones, holgura, distancia de corte, logfactor	Chevychev, coseno	Escalable

	Spectral clustering	Número de clusters, afinidad, matriz diagonalizable, eigenvalores	Chevychev, coseno	Escalabilidad mediano n_samples, pequeñon_clusters
Stats (R)	K-Means	Número de clúster, número de iteraciones	Euclidean, maximum, manhattan, canberra, binary, minkowski	Escalable
clúster (R)	K-Medioids	Número de clúster, métrica distancia	Euclidean, manhattan	Escalable
	CLARA	Número de clúster, métrica distancia, indicador pam, indicador estandarizar, número iteraciones	Euclidean, manhattan	Escalable
	Hierarchical	Matriz distancias, metodología	Euclidean, manhattan	ND
	Fuzzy	Número de clúster, métrica distancia, indicador estandarizar, número iteraciones.	Euclidean, manhattan	Escalable
Mclust (R)	Mclust	Número iteraciones	Gap	ND

DBSCAN (R)	DBSCAN	Radio región vecina, mínimo observaciones por grupo.	Euclidean	Escalabilidad muy grande n_samples, medianon_clusters
SAS	Average linkage	Distancia	Euclidea, dgower	Escalable
	Centroid method	Distancia	Euclidea, dgower	Escalable
	Complete linkage	Distancia	Euclidea, dgower	Escalable en algunos conjuntos de datos
	Density linkage	Distancia	Euclidea, dgower	Escalable en algunos conjuntos de datos
	Expectation Maximization Clustering	Distancia	Euclidea, dgower	No escalable
	Flexible-beta method	Distancia	Euclidea, dgower	No escalable
	McQuitty's similarity analysis	Distancia	Euclidea, dgower	No escalable
	K-Medioids	Distancia	Euclidea, dgower	No escalable
	Single linkage	Distancia	Euclidea, dgower	escalable en algunos conjuntos de datos
	Two-stage density linkage	Distancia	Euclidea, dgower	Escalable en algunos conjuntos de datos
	Ward	Distancia	Euclidea, dgower	Escalable

SPSS	Vinculación intergrupos	Métrica distancia, transformación variables	Distancia euclídea, distancia euclídea al cuadrado, Coseno, Correlación de Pearson, Chebychev, Bloque, Minkowski y Personalizada, Medida de chi-cuadrado y Medida de phicuadrado, diferencia de tamaño, diferencia de configuración, varianza, dispersión, forma, concordancia simple, correlación phi de 4 puntos, lambda, D de Anderberg, dice, hamann, jaccard, kulczynski 1, kulczynski 2, lance y williams, ochiai, rogers y tanimoto, russel y rao, sokal y sneath 1, sokal y sneath 2, sokal y sneath 3, sokal y sneath 4, sokal y sneath 5, Y de yule y Q de yule.	ND

	Vinculación intragrupos	Métrica distancia, Transformación variables	Distancia euclídea, distancia euclídea al cuadrado, Coseno, Correlación de Pearson, Chebychev, Bloque, Minkowski y Personalizada, Medida de chi-cuadrado y Medida de phicuadrado, diferencia de tamaño, diferencia de configuración, varianza, dispersión, forma, concordancia simple, correlación phi de 4 puntos, lambda, D de Anderberg, dice, hamann, jaccard, kulczynski 1, kulczynski 2, lance y williams, ochiai, rogers y tanimoto, russel y rao, sokal y sneath 1, sokal y sneath 2, sokal y sneath 3, sokal y sneath 4, sokal y sneath 5, Y de yule y Q de yule.	ND

	Vecino más próximo (Encadenamiento simple)	Métrica distancia, Transformación variables	Distancia euclídea, distancia euclídea al cuadrado, Coseno, Correlación de Pearson, Chebychev, Bloque, Minkowski y Personalizada, Medida de chi-cuadrado y Medida de phicuadrado, diferencia de tamaño, diferencia de configuración, varianza, dispersión, forma, concordancia simple, correlación phi de 4 puntos, lambda, D de Anderberg, dice, hamann, jaccard, kulczynski 1, kulczynski 2, lance y williams, ochiai, rogers y tanimoto, russel y rao, sokal y sneath 1, sokal y sneath 2, sokal y sneath 3, sokal y sneath 4, sokal y sneath 5, Y de yule y Q de yule.	ND

	Vecino más lejano (Encadenamiento completo)	Métrica distancia, Transformación variables	Distancia euclídea, distancia euclídea al cuadrado, Coseno, Correlación de Pearson, Chebychev, Bloque, Minkowski y Personalizada, Medida de chi-cuadrado y Medida de phicuadrado, diferencia de tamaño, diferencia de configuración, varianza, dispersión, forma, concordancia simple, correlación phi de 4 puntos, lambda, D de Anderberg, dice, hamann, jaccard, kulczynski 1, kulczynski 2, lance y williams, ochiai, rogers y tanimoto, russel y rao, sokal y sneath 1, sokal y sneath 2, sokal y sneath 3, sokal y sneath 4, sokal y sneath 5, Y de yule y Q de yule.	ND

	Hierarchical	Métrica distancia, Transformación variables	Distancia euclídea, distancia euclídea al cuadrado, Coseno, Correlación de Pearson, Chebychev, Bloque, Minkowski y Personalizada, Medida de chi-cuadrado y Medida de phicuadrado, diferencia de tamaño, diferencia de configuración, varianza, dispersión, forma, concordancia simple, correlación phi de 4 puntos, lambda, D de Anderberg, dice, hamann, jaccard, kulczynski 1, kulczynski 2, lance y williams, ochiai, rogers y tanimoto, russel y rao, sokal y sneath 1, sokal y sneath 2, sokal y sneath 3, sokal y sneath 4, sokal y sneath 5, Y de yule y Q de yule.	ND

	K-Medioids	Métrica distancia, Transformación variables	Distancia euclídea, distancia euclídea al cuadrado, Coseno, Correlación de Pearson, Chebychev, Bloque, Minkowski y Personalizada, Medida de chi-cuadrado y Medida de phicuadrado, diferencia de tamaño, diferencia de configuración, varianza, dispersión, forma, concordancia simple, correlación phi de 4 puntos, lambda, D de Anderberg, dice, hamann, jaccard, kulczynski 1, kulczynski 2, lance y williams, ochiai, rogers y tanimoto, russel y rao, sokal y sneath 1, sokal y sneath 2, sokal y sneath 3, sokal y sneath 4, sokal y sneath 5, Y de yule y Q de yule.	ND

	Ward	Métrica distancia, Transformación variables	Distancia euclídea, distancia euclídea al cuadrado, Coseno, Correlación de Pearson, Chebychev, Bloque, Minkowski y Personalizada, Medida de chi-cuadrado y Medida de phicuadrado, diferencia de tamaño, diferencia de configuración, varianza, dispersión, forma, concordancia simple, correlación phi de 4 puntos, lambda, D de Anderberg, dice, hamann, jaccard, kulczynski 1, kulczynski 2, lance y williams, ochiai, rogers y tanimoto, russel y rao, sokal y sneath 1, sokal y sneath 2, sokal y sneath 3, sokal y sneath 4, sokal y sneath 5, Y de yule y Q de yule.	ND
ORANGE	Hierarchical	Modo, número clúster, Número iteraciones	Euclidean, Manhattan, Cosine, Jaccard, Spearman, Pearson, Hamming.Bhattacharyya	ND
	K-Means	Número clúster, scoring, número iteraciones	Euclidean, Manhattan, Cosine, Jaccard, Spearman, Pearson, Hamming.Bhattacharyya	ND
	Louvain Clustering	Métrica distancia, número observaciones, precisión	Euclidean, Manhattan, Cosine, Jaccard, Spearman, Pearson, Hamming.Bhattacharyya	ND
	DBSCAN	Métrica distancia, número clúster	Euclidean, Manhattan, Cosine, Jaccard, Spearman, Pearson, Hamming.Bhattacharyya	ND

	MDS	Métrica distancia, número iteraciones, semilla	Euclidean, Manhattan, Cosine, Jaccard, Spearman, Pearson, Hamming.Bhattacharyya	ND
RAPID MINER	K-Means	Número clúster, número iteraciones	NominalDistance, DiceSimilarity, JaccardSimilarity, KulczynskiSimilarity, RogersTanimotoSimilarity , RussellRaoSimilarity, SimpleMatchingSimilarity, Distancia euclidiana, CanberraDistance, ChebychevDistance, CorrelationSimilarity, CosineSimilarity, DynamicTimeWarpingDistance, InnerProductSimilarity, JaccardSimilarity, MaxProductSimilarity, OverlapSimilarity	ND
	Hierarchical	Modo, métrica distancia, divergencia, kernel	Euclidean,MaxProductSimilarity,JaccardSimilarity, DynamicTimeWarpingDistance, CosineSimilarity, ChebychevDistance, CanberraDistance	ND
	DBSCAN	Número clúster, máximo en grupo, métrica de distancia	Euclidean,MaxProductSimilarity,JaccardSimilarity, DynamicTimeWarpingDistance, CosineSimilarity, ChebychevDistance, CanberraDistance	ND
	Expectation Maximization Clustering	Número clúster, número iteraciones	Euclidean,MaxProductSimilarity,JaccardSimilarity, DynamicTimeWarpingDistance, CosineSimilarity, ChebychevDistance, CanberraDistance	ND

	Random Clustering	Número de clúster	Euclidean,MaxProductSimilarity,JaccardSimilarity, DynamicTimeWarpingDistance, CosineSimilarity, ChebychevDistance, CanberraDistance	ND
	Support Vector Clustering	Mínimo de observaciones en clúster, divergencia	Euclidean,MaxProductSimilarity,JaccardSimilarity, DynamicTimeWarpingDistance, CosineSimilarity, ChebychevDistance, CanberraDistance	ND
	Top Down Clustering	Profundidad máxima clúster	Euclidean,MaxProductSimilarity,JaccardSimilarity, DynamicTimeWarpingDistance, CosineSimilarity, ChebychevDistance, CanberraDistance	ND
	K-Mediods	Número clúster, número iteraciones, métrica distancia, divergencia	Euclidean,MaxProductSimilarity,JaccardSimilarity, DynamicTimeWarpingDistance, CosineSimilarity, ChebychevDistance, CanberraDistance	ND
WEKA	Canopy	Número clúster, métrica de distancia, Densidad minina	Euclídea	ND
	Coweb	Semilla, agudeza, corte	Euclídea	ND
	Expectation Maximization Clustering	Número clúster, número iteraciones	Euclídea	ND
	Fartest First	Número clúster, semilla	Euclídea	ND

	Filtered	Número clúster, semilla	Euclídea	ND
	Hierarchical	Número clúster, Modo, métrica de distancia	Euclídea	ND
	Make Density Based	Número clúster, semilla, media, desviación	Euclídea	ND
	K-Means	Número clúster, número iteraciones	Euclídea	ND
KNIME	K-Means	Número clúster, métrica de distancia, Semilla, número iteraciones, centroides	Euclídea, manhattan, coseno	ND
	K-Medioids	Número clúster, métrica de distancia, Semilla	Euclídea, manhattan, coseno	ND
	Hierarchical	Número clúster, métrica de distancia, modo, tipo de enlace	Euclídea, manhattan, coseno	ND
	OPTICS	Epsilon, métrica distancia, número iteraciones	Euclídea, manhattan, coseno	ND
	Fuzzy	Número clúster, métrica de distancia, número de iteraciones,	Euclídea, manhattan, coseno	ND

		parámetro fuzzy, delta		
	DBSCAN	Epsilon, mínimo de puntos	Euclídea, manhattan, coseno	ND
Matlab	Hierarchical	Número clúster, modo, métrica de distancia	Euclidean, Hamming y Mahalanobis	ND
	K-Means	Número clúster, distancia	Mahalanobis	ND
	Gaussian mixtures	Número clúster	Euclidean, Squared Euclidean,mahalanobis,City block,Minkowski ,Chebychev, cosine, correlation, Hamming distance, jaccard,spearman	ND
	DBSCAN	Epsilon, Mínimo observaciones, distancia	Euclidean,Squared Euclidean, mahalanobis,City block,Minkowski ,Chebychev, cosine, correlation, Hamming distance, jaccard, spearman	ND
	Spectral clustering	Número clúster, Métrica distancia	Euclidean, Squared Euclidean,mahalanobis,City block,Minkowski, Chebychev, cosine, correlation, Hamming distance, jaccard, spearman	ND
	Self-organizing maps	Dimensión, número iteraciones, número clúster	Euclidean, Squared Euclidean,mahalanobis, City block,Minkowski, Chebychev, cosine, correlation, Hamming distance, jaccard, spearman.	ND

	Hidden Markov models	Número de clúster, parámetro convergencia	Euclidean, Squared Euclidean, mahalanobis, City block, Minkowski, Chebychev, cosine, correlation, Hamming distance, jaccard,spearman	ND
	K-Medioids	Número clúster, Métrica distancia	Squared Euclidean	ND

Fuente: Elaboración propia a partir de (Cluster (Mahout Map-Reduce 0.13.0 API), 2017; *Clustering | KNIME*, 2021; *Hierarchical Clustering — Orange Visual Programming 3 documentation*, 2021; RapidMiner GmbH, 2021).

Tabla 3. Algoritmos disponibles en cada librería

Algoritmo/ librería	**Scikit learn (python)**	**mllib (spark)**	**Mahout**	**Cluster (R)**	**SAS**	**SPSS**	**ORANGE**	**RAPID MINER**	**WEKA**	**KNIME**	**Matlab**
K-Means	X	X	X	X	X	X	X	X	X	X	X
Affinity propagation	X										
Mean-shift	X										
Spectral clustering	X		X								X
Ward	X				X	X					
DBSCAN	X			X			X	X		X	X
OPTICS	X									X	
Gaussian mixtures	X	X									X
BIRCH	X										
Power iteration clustering (PIC)		X									
Latent Dirichlet allocation (LDA)		X	X								
Bisecting		X									
Streaming		X	X								
Canopy			X						X		
Fuzzy			X	X						X	
K-Medioids				X		X		X		X	X

CLARA				X							
mclust				X							
average linkage	X		X	X	X	X	X	X	X	X	X
centroid method	X		X	X	X	X	X	X	X	X	X
complete linkage	X		X	X	X	X	X	X	X	X	X
density linkage	X		X	X	X	X	X	X	X	X	X
Expectation Maximization Clustering					X			X	X		
flexible-beta method					X						
McQuitty's similarity analysis					X						
single linkage	X		X	X	X	X	X	X	X	X	X
two-stage density linkage					X						
Vinculación intergrupos						X					
Vinculación intragrupos						X					
Vecino más próximo (Encadenamiento simple)						X					
Vecino más lejano (Encadenamiento completo)						X					
Louvain							X				

Clustering											
MDS							X				
Random Clustering								X			
Support Vector Clustering								X			
Top Down Clustering								X			
Coweb									X		
Fartest First									X		
Filtered									X		
Make Density Based									X		
Self-organizing maps											X
Hidden Markov models											X

Fuente: Elaboración propia a partir de (Cluster (Mahout Map-Reduce 0.13.0 API), 2017; *Clustering | KNIME*, 2021; *Hierarchical Clustering — Orange Visual Programming 3 documentation*, 2021; RapidMiner GmbH, 2021).

Tabla 4. Características principales de las librerías

LIBRERÍAS/ CARACTERÍSTICA	**TIENE GUI**	**APLICABLE BIGDATA**	**REPORTERÍA Y GRÁFICOS ESPECIALIZADOS**
Scikit learn (python)		X	
mlLib(Spark)		X	
Mahout		X	
Clúster (R)			X
SAS	X		X
SPSS	X		
ORANGE	X		X
RAPIDMINER	X		
WEKA	X		X
KNIME	X		
Matlab			X

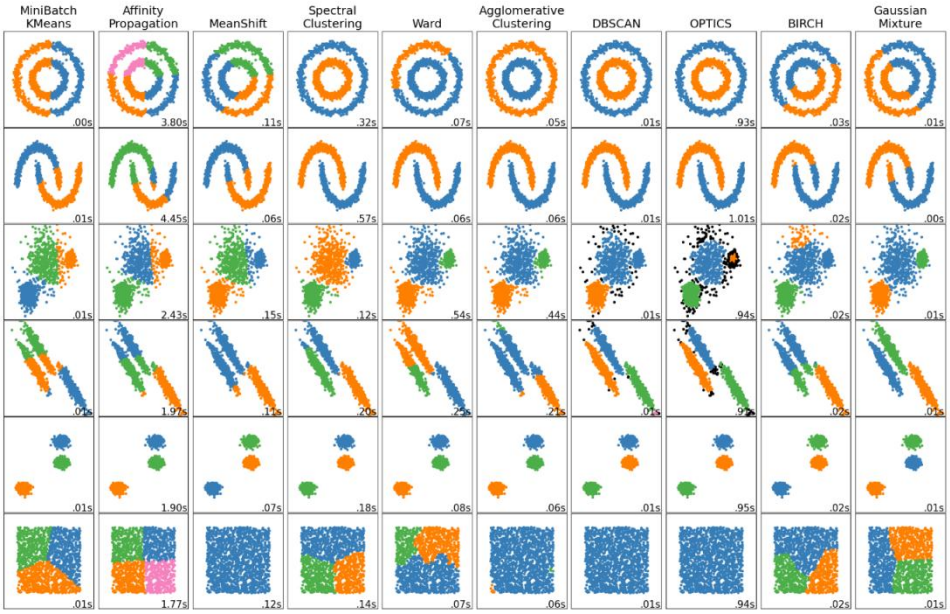
Fuente: Elaboración propia a partir de (Cluster (Mahout Map-Reduce 0.13.0 API), 2017; *Clustering | KNIME*, 2021; *Hierarchical Clustering — Orange Visual Programming 3 documentation*, 2021; RapidMiner GmbH, 2021).

2.6. Problemas no resueltos en las metodologías actuales

El desarrollo de algoritmos de agrupación en clústeres, a través de su historia, se ha venido encontrando una problemática constante, y aún pendiente por resolver y explicar; todavía no existe un método estándar y claro para comparar los algoritmos entre sí, dejando una brecha para estandarizar posibles opciones a la hora de elegir hacer o analizar un clúster.

La interpretación de los distintos clústers formados y su implementación en el entorno original aún no está definido, y a pesar de que cada algoritmo es desarrollado para resolver la misma necesidad, todavía no existe una forma precisa de medir y evaluar la calidad de un algoritmo. De hecho, el proceso de agrupación es impredecible y a veces incluso inconsistente, debido a que el resultado se ve afectado por la subjetividad del usuario al escoger el algoritmo y al parametrizarlo, ya que para un mismo conjunto de datos se puede generar diferentes agrupaciones, según el algoritmo usado y su respectiva parametrización (Gelbard, Goldman, & Spiegler 2007). Esta problemática se puede evidenciar en la siguiente imagen:

Ilustración I. Resultados de algoritmos según una distribución de datos



Fuente: Tomado de: (Scikit-learn, S.F.)

Dada la problemática de la dificultad con la que se encuentra el usuario para seleccionar el algoritmo adecuado, y teniendo en cuenta la gran cantidad de algoritmos de agrupamiento que encontramos; se pueden clasificarlos de la siguiente manera (Meilă, & Heckerman, 2001):

- **Hierarchical (Jerárquica):** Construye grupos a través de jerarquías con las medidas de similitud.
- **Partitioning:** Construye grupos a través de medidas de similitud (Media, Mediana, Moda, etc).
- **Density-based:** Construye clúster basado en espacio de datos como una región contigua de alta densidad de puntos, separada de otros clústeres similares por regiones contiguas de densidad de puntos baja. Los puntos de datos en las regiones de separación de baja densidad de puntos generalmente se consideran ruido / valores atípicos. (dbscan, optics).
- **Model-based:** Permite identificar clusters en función de su forma y estructura en lugar de la proximidad entre puntos de datos. diseñados para modelar una distribución desconocida como una mezcla de distribuciones más simples, a veces llamadas distribuciones de base.
- **Grid-based**: utiliza una estructura de datos de cuadrícula de resolución múltiple. Forma la estructura de datos de la cuadrícula dividiendo el espacio de datos en varias celdas y realiza la agrupación en las cuadrículas. La agrupación depende del número de celdas de la cuadrícula e independiente del número de objetos de datos. reducción significativa de la complejidad computacional, especialmente para agrupar conjuntos de datos muy grandes. El enfoque de agrupamiento basado en cuadrículas se diferencia de los algoritmos de agrupamiento convencionales en que no se ocupa de los puntos de datos sino del espacio de valores que rodea los puntos de datos.

  Esta clasificación ayuda a identificar qué tipo de algoritmo puede ser más conveniente, Dentro de cada clasificación se han desarrollado diferentes algoritmos que mejoran problemas existentes o se usan para tipos de bases de datos específicos, a continuación, se mencionan algunos encontrados en la literatura, con sus características y pertinencia de uso según el caso:

  - **Optics**: No sólo presenta información de agrupación "tradicional" (por ejemplo, puntos representativos, agrupaciones de formas arbitrarias), sino agrupación basada en densidad y la estructura de agrupación intrínseca. Se desarrolló con el objetivo de ofrecer información adicional sobre la distribución y correlación de los datos, con respecto a los algoritmos existentes en la época. No es factible aplicarlo a una base de datos que contiene varios millones de objetos [21].
  - **DBSCAN:** Realiza agrupaciones basadas en la densidad, está diseñado para descubrir agrupaciones de forma arbitraria, ya que está basado en cuadrículas. Se desarrolla con el fin de simplificar el proceso de análisis para datos de baja dimensión, ya que sólo requiere un parámetro de entrada y tiene un tiempo de complejidad menor [21].
  - **BIRCH:** Agrupamiento arbitrario, razonablemente rápido. Se puede utilizar como una alternativa al muestreo de datos para mejorar la escalabilidad de algoritmos de agrupamiento. En el momento se consideraba el mejor método de agrupación disponible para bases de datos muy grandes, ya que permite que un gran problema de agrupamiento sea manejable al concentrarse en porciones densamente ocupadas y resumen compacto.

Es muy usado en procesamiento de imágenes (Ankerst, Breunig, Kriegel, & Sander, 1999; Zhang, Ramakrishnan, & Livny, 1996)

- **KMEANS:** Desde su creación ha sido muy usado debido a que es computacionalmente económico y por lo general, los usuarios lo consideran de fácil uso (Maheswaran, Jayarajan, & Joseph, 2013; MacQueen, 1967).
- **K MEDOIDS**: Ayuda a purificar el agrupamiento de datos atípicos. Funciona de manera ineficiente para un gran conjunto de datos debido a su complejidad (Kaufman, L., & Rousseeuw, 1990).
- **CLARA:** Toma varias muestras y agrupa todo el conjunto de datos varias veces para representar distribuciones en muestras pequeñas por lo que tiene un mejor rendimiento con gran conjunto de datos, pero no muy bueno con una cantidad de clústers alta (Dane, & Kateman, 1993).
- **K PROTOTIPOS:** Comprime los datos en prototipos con una mínima participación del usuario. Esto disminuye la necesidad de habilidad estadística y computacional de los usuarios finales (Äyrämö, & Kärkkäinen, 2006). Integra los procesos k-means y k-modes para agrupar datos con combinaciones numéricas y valores categóricos. Se define como una medida de disimilitud que toma en cuenta tanto atributos numéricos como categóricos (Huang, 1998). CLARA agrupa un gran conjunto de datos basado en muestras, mientras que los K-PROTOTIPOS funcionan directamente en todo el conjunto de datos. El algoritmo de k-prototipos no tiene limitación cuando el tamaño de la muestra excede un cierto rango, digamos miles de objetos -es más rápido que k medias algoritmo, porque necesita menos iteraciones para converger.

Una vez seleccionado el algoritmo, se pueden encontrar formas tradicionales de evaluar las estructuras de agrupamiento, comparándolas con otros esquemas producidos por el mismo método, pero con diferentes valores de parámetros, arrojando un supuesto de qué tan acertada fue la agrupación por los algoritmos elegidos y utilizados. A continuación, se definirán las metodologías más comunes para evaluar agrupaciones (Aliguliyev, 2009):

- **Revisión interna:** Homogeneidad o dispersión dentro de los conglomerados y entre las separaciones de conglomerados.
- **Revisión Externa:** La validación se refiere a comparar una solución de agrupación en clústeres con una agrupación verdadera.
- **Criterios relativos:** Para la investigación de la calidad de los conglomerados. La idea básica es la evaluación de una estructura de agrupamiento comparándola con otros esquemas de agrupamiento producidos por el mismo método, pero con diferentes valores de parámetros de entrada.

Estos métodos no dan una respuesta específica de cuál agrupación es la mejor, sin embargo, brinda al usuario una información más completa y con criterios estructurados, que le permita realizar una toma de decisiones más consciente de acuerdo con su contexto.

2.7. Metodología propuesta

Para el desarrollo de un prototipo de librería que permita el análisis automático de clústers, mediante métodos identificados en el estado del arte y que facilite las decisiones de modelado por parte del usuario, se partió de un diseño metodológico mixto involucrando una parte exploratoria y otra de desarrollo.

En primer lugar, se realizó una revisión bibliográfica, donde se identificaron las metodologías más utilizadas para la detección de grupos; esto por medio de bases de datos, libros y artículos científicos.

En segundo lugar, se identifica cada parte teniendo en cuenta todo el proceso de clustering: lectura de datos, análisis descriptivo de datos (gráficas y resúmenes), preparación de variables, selección de distancia, selección cantidad de clusters, selección de algoritmo de clúster, selección cantidad de clusters, selección del algoritmo, implementación algoritmo, análisis de resultados, exportar resultados.

En tercer lugar, se realiza una herramienta que contiene los principales beneficios de las herramientas ya existentes, como la implementación de algunos de los algoritmos más usados como: kmeans, kmedoids, kprototypes, entre otros; disponiendo de una interfaz gráfica amigable, que permita la interacción con cualquier tipo de usuario sin importar sus conocimientos en programación o estadística. Adicionalmente, se agrega un componente que se divide en dos partes: la primera, es la sugerencia de parámetros y algoritmo a usarse a través del análisis de datos ingresados, reduciendo la subjetividad al momento realizar el proceso de clustering; y la segunda, es la interpretación de resultados facilitando la comprensión y el análisis de los datos proporcionados por el algoritmo implementado.

Por último, se realizaron las pruebas de validación, permitiendo comprobar la funcionalidad de la librería.

2.8. Descripción del software

A continuación, se realizará una descripción detallada del prototipo de librería desarrollado, que permite el análisis automático de clústers, mediante las metodologías identificadas y que facilitan las decisiones de modelado por parte del usuario.

2.8.1. Arquitectura de software

El software está implementado en Python 3.7.1 (Tutorial de Python, 2021) a través de un jupyter notebook (The Jupyter Notebook, 2021) lo que facilita la interacción con el usuario y el conocimiento del código. Se utilizan varias librerías para manejo de conjunto de datos y

operaciones numéricas y matriciales como numpy (NumPy, 2021) y pandas (Pandas, 2021), Librerías que facilitan la interacción con el usuario y la visualización de datos como matplotlib (Matplotlib, 2012), yellowbrick (Yellowbrick, 2021), ipwidgets (ipywidgets, 2021), entre otras; y librerías para el manejo de errores y excepciones como sys (sys, 2021), time (time, 2021) y warning (warnings, 2021).

Adicionalmente, se implementan diferentes algoritmos que permiten analizar los datos y explorar el mejor modelo posible con ayuda de librerías como scikit-learn (Scikit-learn, S.F.).

2.8.2. Funcionalidades del software

El software permite de una manera gráfica, cargar un conjunto de datos y a través de un análisis descriptivo sencillo poder tomar decisiones en el momento de iniciar agrupación a través de técnicas de clusterización. El software provee sugerencias de los parámetros a utilizar y el algoritmo adecuado según el conjunto de datos y el usuario podrá realizar el proceso tomando las sugerencias o realizando modificaciones hasta llegar al resultado deseado. Adicionalmente se presentan los resultados de manera gráfica permitiendo una interpretación fácil e intuitiva de estos, como también permitiendo al usuario exportar sus resultados.

2.8.3. Ejemplos ilustrativos

Dentro de la documentación del software (Correa, 2021), la cual se encuentra en el repositorio en línea, dentro de la carpeta docs, se incluye un manual de uso para el usuario con varios ejemplos ilustrativos y casos de uso.

2.8.4. Beneficios y contribuciones de la metodología propuesta

En este trabajo, se han analizado diferentes herramientas que permiten el análisis automático de clústers, y en la (Tabla 4) se puede ver cuáles de las principales herramientas existentes tienen las funcionalidades de GUI, aplicabilidad de Big Data, reportería y gráficos especializados; todas presentan alguna o máximo dos de las funciones anteriores.

En contraste con lo anterior, la solución desarrollada en este trabajo tiene como ventaja que presenta las tres funcionalidades, lo cual genera facilidades para la unificación de necesidades a la hora de construir clústers. Adicionalmente, al ser desarrollada en Python, permite usar su amplia gama de paquetes para todo tipo de técnicas de clusterización, presenta la posibilidad de generar integraciones futuras con spark, de manera que se pueda trabajar grandes volúmenes de datos de una forma óptima. Así mismo, la solución aprovecha los paquetes de visualización y análisis descriptivos de datos y también adiciona una interfaz gráfica que permite facilidad de interacción con cualquier usuario.

En línea con lo anterior, la solución analiza cada tipo de dato para ofrecer la sugerencia de parametrización más conveniente en la clusterización, lo que permite disminuir la subjetividad o los errores por desconocimiento de la técnica a implementar; además, presenta una interpretación de los datos intuitiva para cualquier usuario y es gratuita.

3. Impacto

Este trabajo, permitió el desarrollo de un prototipo de librería para el análisis automático de clústers mediante diferentes metodologías, que facilita las decisiones de modelado por parte del usuario.

Así mismo con esta solución, se pretende disminuir el error en el tratamiento de datos al realizarse un análisis previo y sugerir las opciones más adecuadas según las características de los datos, logrando así mejores prácticas y un mejor agrupamiento de datos.

Adicionalmente, se puede mejorar en gran medida la toma de decisiones en diferentes disciplinas que, en su conocimiento de base, no incluyen conceptos de programación o estadística de manera profunda, por lo que la automatización en la parametrización permite ahorros en tiempos y en errores.

Para futuros estudios, se sugeriría la implementación de otras herramientas y otros algoritmos que complementen el análisis clúster con análisis estadísticos adicionales que permitan la toma de decisiones de los usuarios.

4. Conclusiones

La elaboración de este trabajo, permitió desarrollar un prototipo de librería que permite análisis automático de clusters, mediante metodologías del estado del arte y facilitan las decisiones de modelado por parte del usuario.

Después de analizar y estudiar las librerías y herramientas actuales de agrupación de clúster, se identifican tres funcionalidades principales, las cuales ninguna de las soluciones existentes presenta de manera integral.

Se logró desarrollar una nueva librería que integra y contiene las tres funcionalidades principales para la agrupación de Clúster (GUI, aplicabilidad de Big Data, reportería y gráficos especializados).

Se espera que la herramienta desarrollada pueda facilitar la toma de decisiones de los usuarios que la utilicen en las diferentes aplicaciones que los estudios clúster impactan.

5. Referencias

Aguilar, L. J. (2016). Big Data, Análisis de grandes volúmenes de datos en organizaciones. Alfaomega Grupo Editor.

Aldenderfer, M. S., & Blashfield, R. K. (1984). A review of clustering methods. Cluster analysis, 33-61.

Aliguliyev, R. M. (2009). Performance evaluation of density-based clustering methods. Information Sciences, 179(20), 3583-3602.

Aluja, T. (2001). La minería de datos, entre la estadística y la inteligencia artificial. Qüestió: quaderns d'estadística i investigació operativa, 25(3), 479-498.

Anderberg, M. R. (1973). Cluster Analysis for applications. Academic Press. New York and London.

Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. ACM Sigmod record, 28(2), 49-60.

Ashenden,A., Ward-Dutton, N., & Wentworth, C., (2016). La nueva tendencia de automatización: Machine Learning y más. MWD Advisors. Disponible en: https://www.ibm.com/downloads/cas/M1PG1J23.

Äyrämö, S., & Kärkkäinen, T. (2006). Introduction to partitioning-based clustering methods with a robust example. Reports of the Department of Mathematical Information Technology. Series C, Software engineering and computational intelligence, (1/2006).

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1), 208-221.

Chojnacki, A., Dai, C., Farahi, A., Shi, G., Webb, J., Zhang, D.T., Abernethy, J., Schwartz, E., (2017). A Data Science Approach to Understanding Residential Water Contamination in Flint, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’17. ACM, New York, NY, USA, pp. 1407–1416. https://doi.org/10.1145/3097983.3098078

Chou, Y. L., & Armer, V. A. (1977). Análisis estadístico (No. 04; RMD, HA29 C4 1977.). Interamericana.

Cleveland, W.S., 2001. Data Science: an Action Plan for Expanding the Technical Areas of the Field of Statistics. Int. Stat. Rev. 69, 21–26. https://doi.org/10.1111/j.1751-5823.2001.tb00477.x

Cluster (Mahout Map-Reduce 0.13.0 API). (2017, April 14). Apache.org. https://mahout.apache.org/docs/0.13.0/api/docs/mahout-mr/org/apache/mahout/clustering/Cluster.html

Clustering | KNIME. (2021). KNIME. https://www.knime.com/nodeguide/analytics/clustering

Dhillon, I. S., Guan, Y., & Kulis, B. (2004, August). Kernel k-means: spectral clustering and normalized cuts. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 551-556). ACM.

Correa, M., (2021, October 12). TDGMarisolCorreaHenao/docs at main · marcorhe/TDGMarisolCorreaHenao. GitHub. https://github.com/marcorhe/TDGMarisolCorreaHenao/tree/main/docs

Dane, A. D., & Kateman, G. (1993). On k-medoid clustering of large data sets with the aid of a genetic algorithm: Background, feasibility and comparison. Analytica Chimica Acta, 282, 647–669. 2009 a simple

Díaz, M., León, Á., Alvin, H., & Díaz Mora, M. E. (2016). Introducción al análisis estadístico multivariado aplicado. Experiencia y casos en el Caribe colombiano. Universidad del Norte.

Dubes, R., & Jain, A. K. (1979). Validity studies in clustering methodologies. Pattern recognition, 11(4), 235-254.

Eluri, V. R., Ramesh, M., Al-Jabri, A. S. M., & Jane, M. (2016, March). A comparative study of various clustering techniques on big data sets using Apache Mahout. In 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC) (pp. 1-4). IEEE.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

Fernández, S. F., Sánchez, J. M. C., Córdoba, A., & Largo, A. C. (2002). Estadística descriptiva. Esic Editorial.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In Advances in neural information processing systems (pp. 2962-2970).

Gelbard, R., Goldman, O., & Spiegler, I. (2007). Investigating diversity of clustering methods: An empirical comparison. Data & Knowledge Engineering, 63(1), 155-166.

Gómez-Skarmeta, A. F., Delgado, M., & Vila, M. A. (1999). About the use of fuzzy clustering techniques for fuzzy model identification. Fuzzy sets and systems, 106(2), 179-188.

Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A., (2014). Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. Int. J. Prod. Econ. 154, 72–80. https://doi.org/10.1016/j.ijpe.2014.04.018.

Hierarchical Clustering — Orange Visual Programming 3 documentation. (2021). Readthedocs.io. https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery, 2(3), 283-304.

ipywidgets — Jupyter Widgets 7.6.5 documentation. (2021). Readthedocs.io. https://ipywidgets.readthedocs.io/en/stable/

Ji, J., Bai, T., Zhou, C., Ma, C., & Wang, Z. (2013). An improved k-prototypes clustering algorithm for mixed numeric and categorical data. Neurocomputing, 120, 590-596.
SAS Institute. (2012). SAS/OR 9.3 User's Guide: Mathematical Programming Examples. SAS institute.

Kaufman, L., & Rousseeuw, P. J. (1990). Finding groups in data: An introduction to clúster analysis. New York: Wiley 2009 a simple.

López, C. P. (2007). Minería de datos: técnicas y herramientas. Editorial Paraninfo.

Lückeheide, S., Velásquez, J. D., & Cerda, L. (2007). Segmentación de los contribuyentes que declaran iva aplicando herramientas de clustering. Revista de Ingeniería de Sistemas, 21, 87-110.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (Vol. 1, pp. 281–297). Berkeley: University of California Press. 2009 a simple.

Maheswaran, G., Jayarajan, P., Jose, J., & Joseph, J. (2013). K Means Clustering Algorithms: A Comparitive Study.

Matplotlib: Python plotting — Matplotlib 3.4.3 documentation. (2012). Matplotlib.org. https://matplotlib.org/

Meilă, M., & Heckerman, D. (2001). An experimental comparison of model-based clustering methods. Machine learning, 42(1), 9-29.

Morán, L. L., & Alonso, J. H. (2009). Estadística descriptiva. Ediciones Académicas.

Müllner, D. (2013). fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python. Journal of Statistical Software, 53(9), 1-18.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems (pp. 849-856).

NumPy. (2021). Numpy.org. https://numpy.org/

Pandas - Python Data Analysis Library. (2021). Pydata.org. https://pandas.pydata.org/

Park, H. S., & Jun, C. H. (2009). A simple and fast algorithm for K-medoids clustering. Expert systems with applications, 36(2), 3336-3341.

Peña, D. (2002). Análisis de datos multivariantes (Vol. 24). Madrid: McGraw-hill.

Provost, F., Fawcett, T., (2013). Data Science for Business: What you need to know about data mining and data-analytic thinking. O'Reilly Media, Inc.

RapidMiner GmbH. (2021). k-Means - RapidMiner Documentation. Rapidminer.com. https://docs.rapidminer.com/latest/studio/operators/modeling/segmentation/k_means.html

Reynolds, A. P., Richards, G., & Rayward-Smith, V. J. (2004, August). The application of k-medoids and pam to the clustering of rules. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 173-178). Springer, Berlin, Heidelberg.

Rodrigo, J.A. (2020). Clustering con Python. Disponible en https://www.cienciadedatos.net/documentos/py20-clustering-con-python.html [10-06-2021].

Ross, S. M. (2007). Introducción a la estadística. Reverté.

Sangüesa Solé, Ramón (coord.) (2000). Data mining: una introducción. Barcelona: Universitat Oberta de Catalunya.

Santana, Ó. F. (1991). El análisis de clúster: aplicación, interpretación y validación. Papers: revista de sociologia, (37), 65-76.

Scikit-learn, (S.F.). Scikit-learn: Clustering. Disponible en: https://scikit-learn.org/stable/modules/clustering.html.

SAS/STAT Cluster Analysis Procedures. (2018, November 20). Sas.com. https://support.sas.com/rnd/app/stat/procedures/ClusterAnalysis.html

Shen, J., Hao, X., Liang, Z., Liu, Y., Wang, W., & Shao, L. (2016). Real-time superpixel segmentation by DBSCAN clustering algorithm. IEEE Transactions on Image Processing, 25(12), 5933-5942.

sys — Parámetros y funciones específicos del sistema — documentación de Python - 3.10.0. (2021). Python.org. https://docs.python.org/es/3.10/library/sys.html

The Jupyter Notebook — Jupyter Notebook 6.4.5 documentation. (2021). Readthedocs.io. https://jupyter-notebook.readthedocs.io/en/stable/

time — Time access and conversions — Python 3.10.0 documentation. (2021). Python.org. https://docs.python.org/3/library/time.html

Tutorial de Python — documentación de Python - 3.10.0. (2021). Python.org. https://docs.python.org/es/3/tutorial/

Uriel, E., & Aldás, J. (2005). Análisis multivariante aplicado. 1ª. Edición. Thomson. Madrid.

Van der Aalst, W.M., 2016. *Process mining: data science in action.* Springer.

Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing, 17(4), 395-416.

Wang, K., Zhang, J., Li, D., Zhang, X., & Guo, T. (2008). Adaptive affinity propagation clustering. arXiv preprint arXiv:0805.1096.

warnings — Warning control — Python 3.10.0 documentation. (2021). Python.org. https://docs.python.org/3/library/warnings.html

Yellowbrick: Machine Learning Visualization — Yellowbrick v1.3.post1 documentation. (2021). Scikit-Yb.org. https://www.scikit-yb.org/en/latest/

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. ACM sigmod record, 25(2), 103-114.

Zelnik-Manor, L., & Perona, P. (2005). Self-tuning spectral clustering. Advances in neural information processing systems (pp. 1601-1608).