UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

DESARROLLO Y GESTIÓN DE SOFTWARE

Extracción de Conocimiento en Bases de Datos

IV.1. Algoritmos de agrupación y reducción de dimensionalidad

DOCENTE:

ING. LUIS ENRIQUE MASCOTE CANO

PRESENTAN:

DARON TARÍN GONZÁLEZ

ÁNGEL RICARDO CHÁVEZ ZARAGOZA

MILDRED VILLASEÑOR RUIZ

GRUPO:

IDGS91N

Chihuahua, Chih., 30 de noviembre de 2025

Contenido

Figuras y tablas

Objetivo

Identificar y describir los principales algoritmos de agrupación (clustering) y de reducción de dimensionalidad, así como ejemplificar su uso mediante explicaciones conceptuales y ejemplos aplicados.

Introducción

En extracción de conocimiento, los datos suelen ser abundantes y complejos, lo cual dificulta su interpretación directa. Por ello, dos técnicas fundamentales en análisis exploratorio son el **clustering**, que permite agrupar instancias similares sin necesidad de etiquetas, y la **reducción de dimensionalidad**, que simplifica conjuntos de datos con muchas variables manteniendo su estructura esencial. El clustering ayuda a descubrir patrones, segmentos y comportamientos ocultos; mientras que la reducción de dimensionalidad mejora la visualización, disminuye el ruido, reduce costos computacionales y facilita modelos posteriores de machine learning. Estas técnicas son esenciales en áreas como marketing, bioinformática, visión por computadora y minería de datos. Este reporte describe tres algoritmos de agrupación —K-means, Clustering jerárquico y DBSCAN— y dos métodos de reducción de dimensionalidad: LDA y Autoencoders.

1. Algoritmos de agrupación (Clustering)

1.1 K-means

El algoritmo K-means es uno de los más utilizados en agrupación debido a su simplicidad y eficiencia. Su objetivo es dividir un conjunto de datos en *k* grupos, donde cada clúster se representa mediante un centroide. El proceso consiste en inicializar los centroides, asignar cada punto al centroide más cercano, recalcular los centroides como el promedio de los puntos asignados y repetir hasta alcanzar convergencia.

Principio de funcionamiento

- Inicializar *k* centroides.
- Asignar cada punto al centroide más cercano usando distancia euclidiana.
- Recalcular centroides con el promedio de los puntos del clúster.
- Repetir hasta que los centroides no cambien significativamente.

Parámetros clave

- Número de clústeres (*k*).
- Método de inicialización (aleatorio o k-means++).
- Métrica de distancia.
- Número máximo de iteraciones.

Ventajas

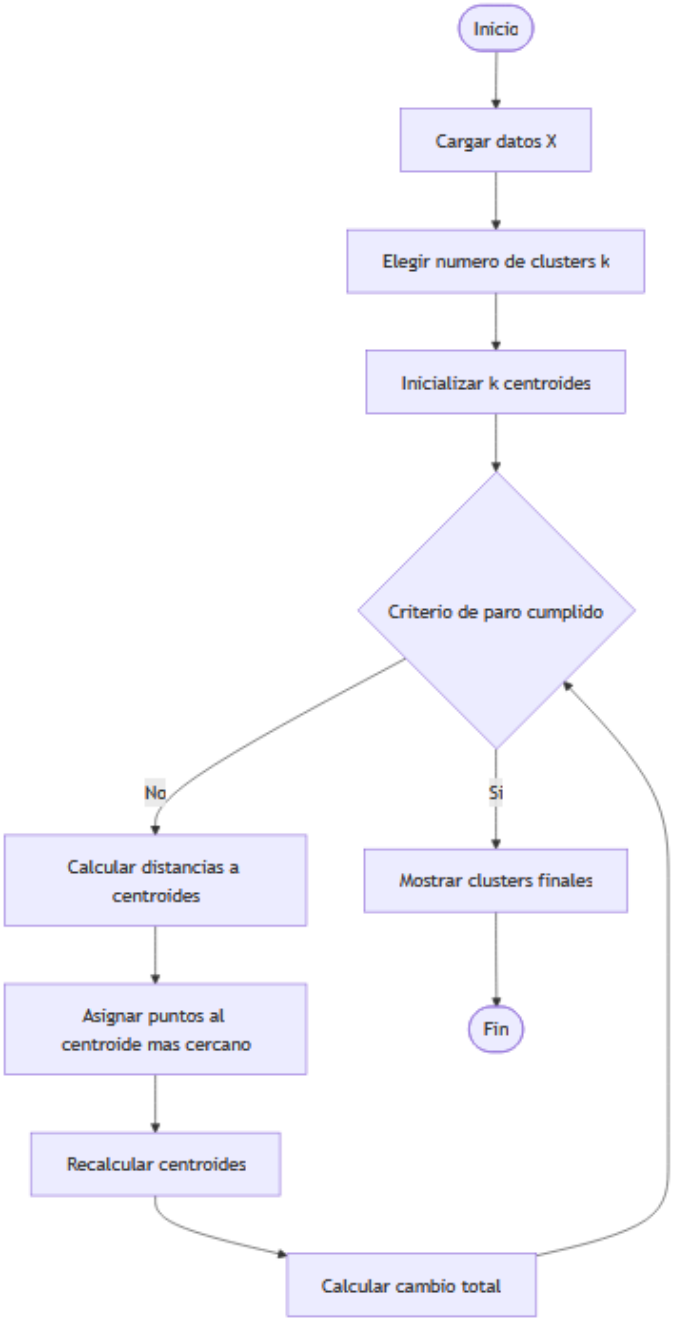
- Muy rápido y escalable.
- Fácil de implementar.
- Funciona bien con clústeres esféricos.

Limitaciones

- Requiere definir *k* previamente.
- Sensible a outliers.
- No funciona bien con clústeres de densidad variable o formas irregulares.

Figura 1

Diagrama de flujo del algoritmo K-means.



1.2 Clustering jerárquico aglomerativo

El clustering jerárquico aglomerativo construye una estructura en forma de árbol llamada dendrograma, mostrando cómo los datos se fusionan progresivamente desde clústeres individuales hasta un solo grupo general. A diferencia de métodos como K-means, no requiere fijar el número de clústeres al inicio.

Principio de funcionamiento

- Cada punto inicia como un clúster individual.
- Se calculan distancias entre clústeres.
- Se fusionan los dos clústeres más cercanos.
- El proceso se repite hasta formar un único clúster o alcanzar un número deseado.

Parámetros clave

- Método de enlace:
    - *single*: distancia mínima.
    - *complete*: distancia máxima.
    - *average*: distancia promedio.
- Métrica de distancia.
- Nivel al que se corta el dendrograma.

Ventajas

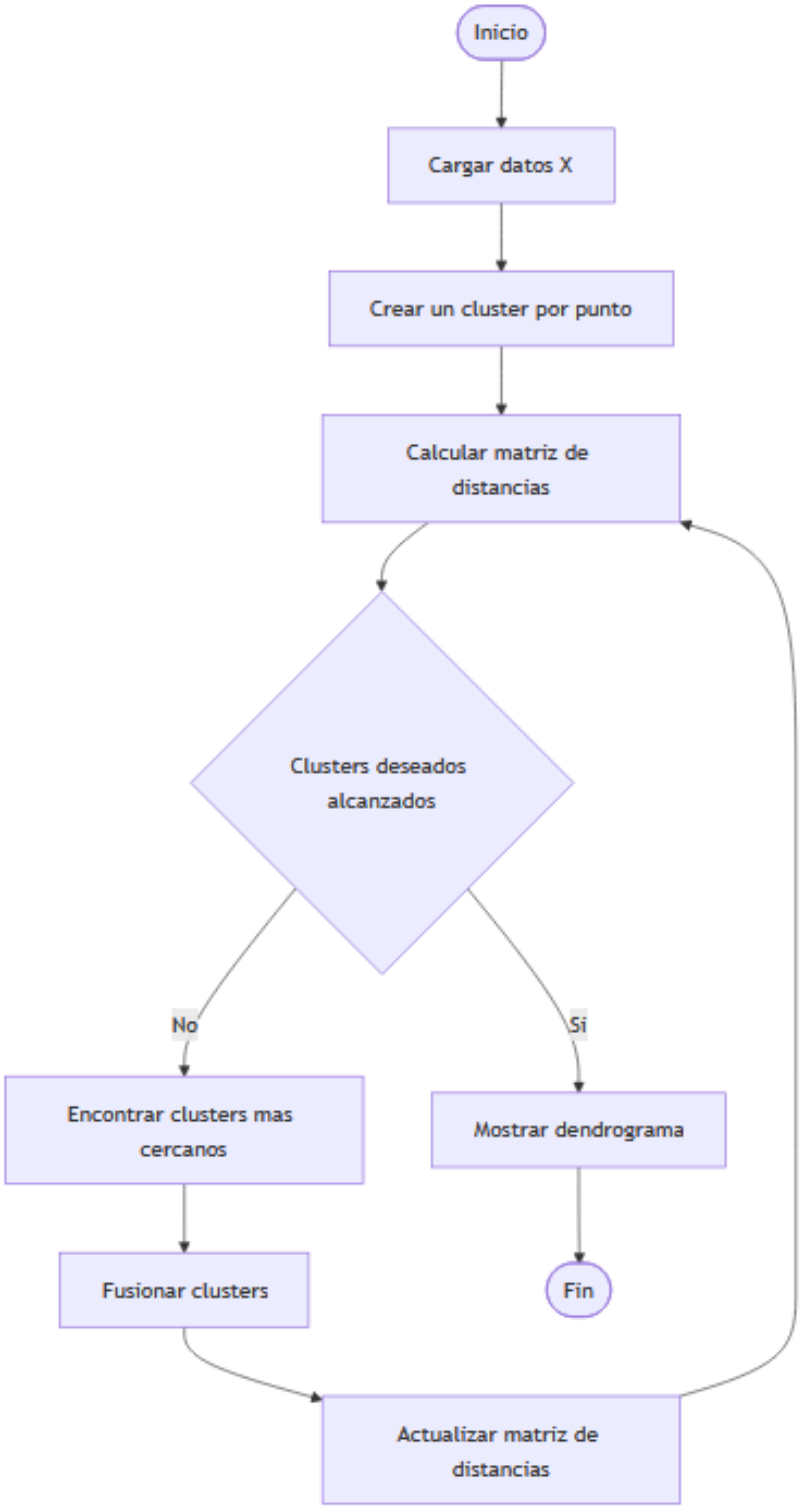
- No requiere definir *k* desde el inicio.
- Ofrece una representación visual clara mediante dendrogramas.
- Funciona bien para análisis exploratorio.

Limitaciones

- Costoso computacionalmente para conjuntos grandes.
- Las fusiones no pueden revertirse.
- Sensible al método de enlace utilizado.

Figura 2

Diagrama de flujo del clustering jerárquico aglomerativo.



1.3 DBSCAN

DBSCAN agrupa datos según la densidad local de puntos. Identifica como clústeres aquellas regiones donde existan suficientes puntos cercanos entre sí. También detecta ruido de manera natural, diferenciándose de algoritmos como K-means.

Principio de funcionamiento

- Usa dos parámetros: *eps* (radio máximo) y *minPts* (mínimo de vecinos).
- Los puntos pueden ser: núcleo, frontera o ruido.
- Los clústeres se expanden a partir de puntos núcleo conectados por densidad.

Parámetros clave

- eps (radio).
- minPts (densidad mínima).
- Métrica de distancia.

Ventajas

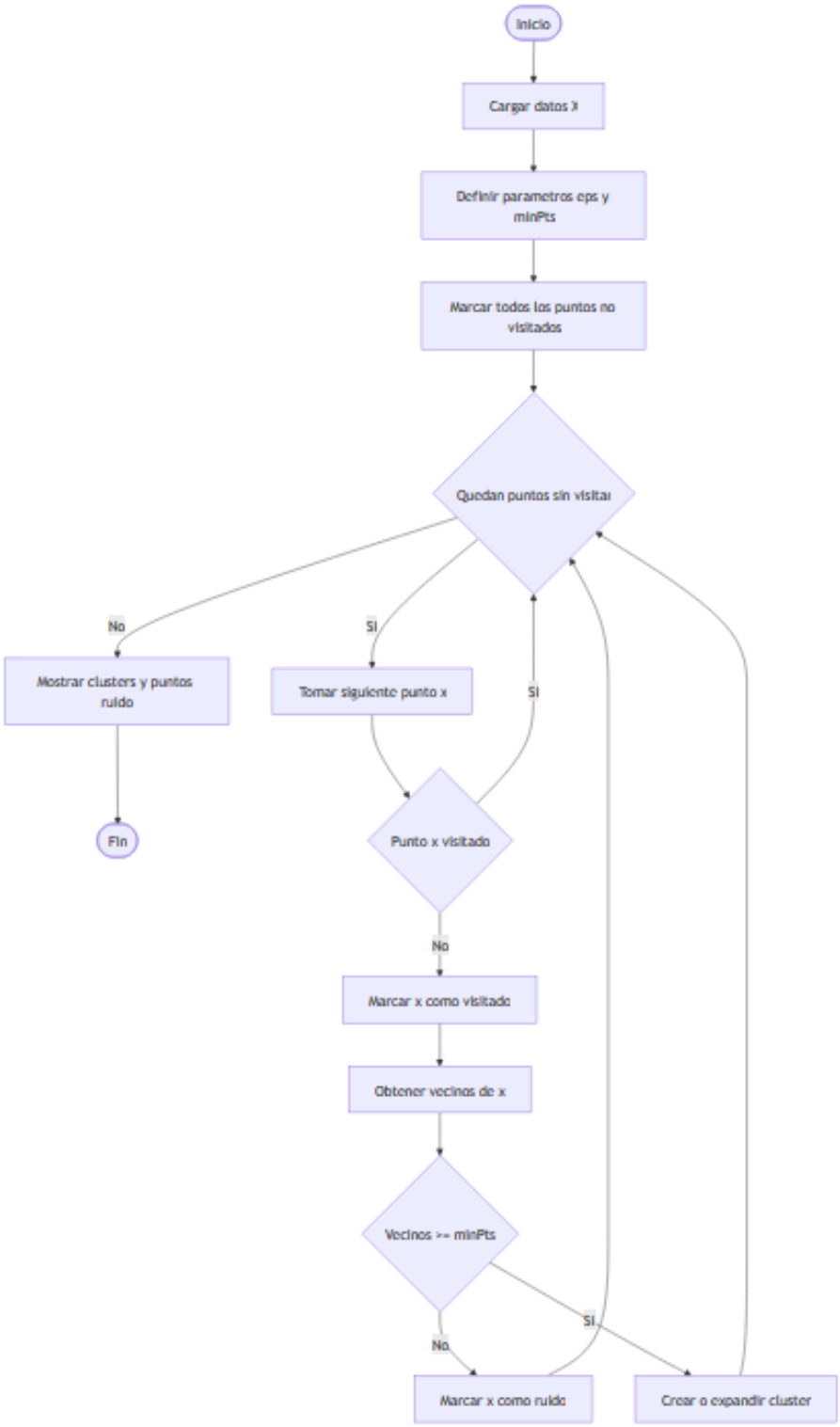
- Detecta clústeres de forma arbitraria.
- Bueno con ruido y valores atípicos.
- No requiere definir *k*.

Limitaciones

- Difícil elegir parámetros óptimos.
- Fallos si la densidad varía entre regiones.

Figura 3

Diagrama de flujo del algoritmo DBSCAN.



2. Algoritmos de reducción de dimensionalidad

2.1 Análisis Discriminante Lineal (LDA)

El LDA es un método supervisado que busca proyectar los datos en un espacio de menor dimensión maximizando la separación entre clases. Considera las medias por clase y optimiza la razón entre la varianza entre clases y la varianza dentro de las clases.

Fundamento conceptual

- Calcula matrices de dispersión entre clases y dentro de clases.
- Optimiza la proyección mediante autovectores.
- Produce componentes que mejor discriminan clases.

Parámetros clave

- Número de componentes discriminantes.
- Método de descomposición (eigenvalue).

Ventajas

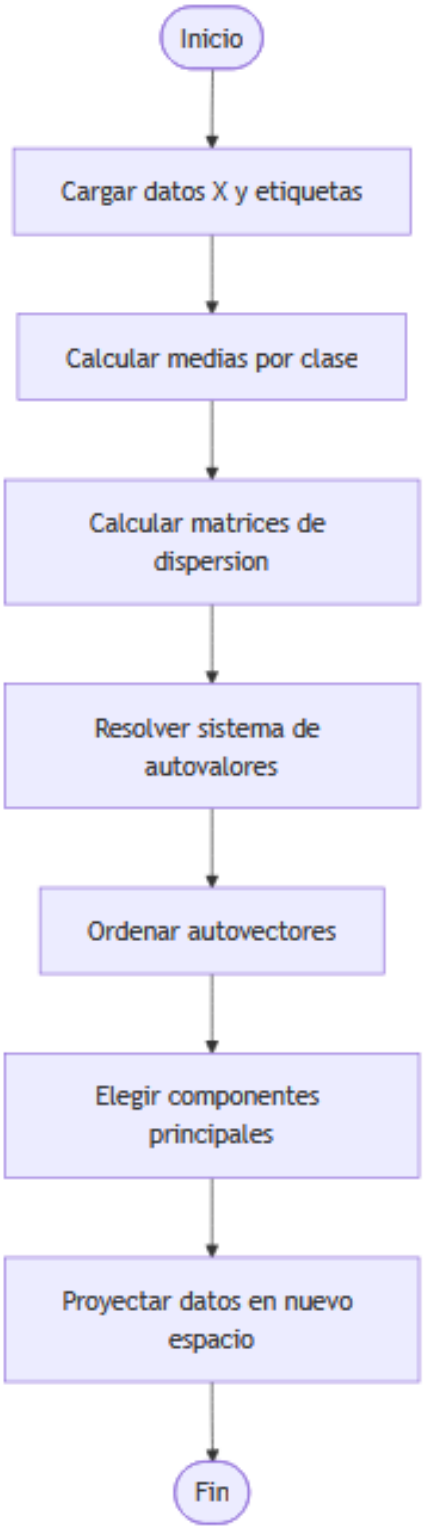
- Muy útil para clasificación.
- Reduce dimensionalidad de forma interpretativa.
- Computacionalmente eficiente.

Limitaciones

- Supone separación lineal entre clases.
- Sensible a datos no balanceados.

Figura 4

Diagrama de flujo del Análisis Discriminante Lineal (LDA).



2.2 Autoencoders

Los autoencoders son redes neuronales diseñadas para aprender representaciones comprimidas de los datos. Constan de un codificador que reduce la dimensión y un decodificador que reconstruye la entrada.

Fundamento conceptual

- La red aprende a minimizar el error entre entrada y salida.
- Obliga a la red a aprender patrones esenciales.
- Permite representaciones no lineales.

Parámetros clave

- Tamaño de la capa latente.
- Número de capas.
- Funciones de activación.
- Épocas y optimizador.

Ventajas

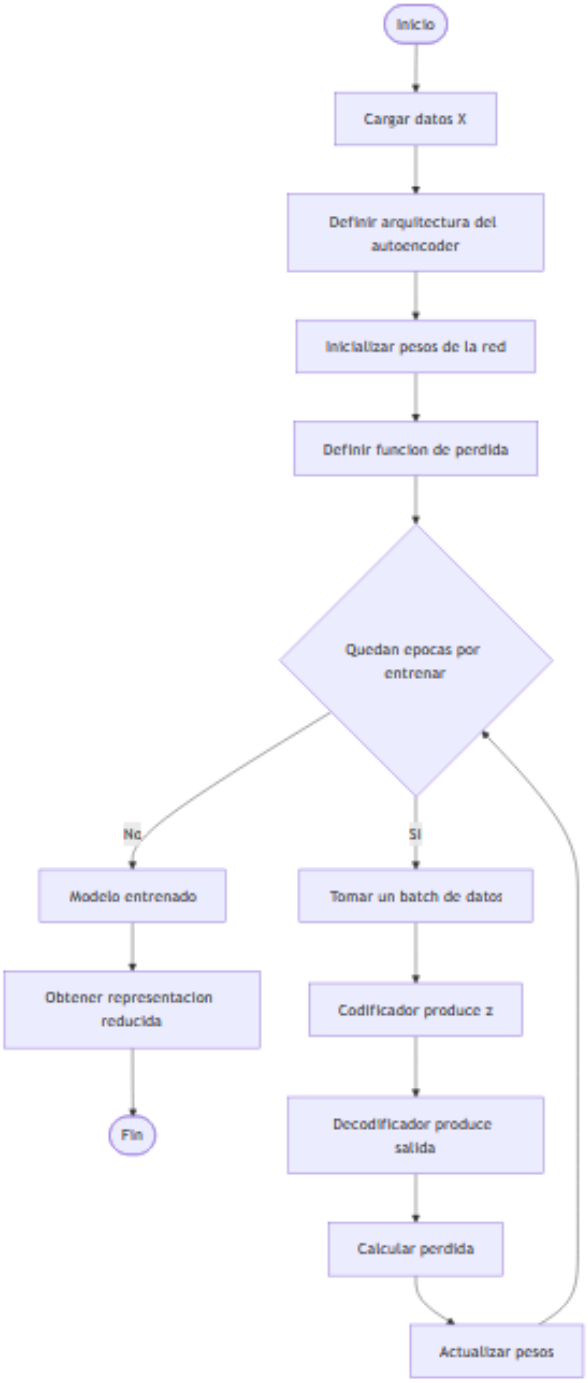
- Capturan relaciones no lineales.
- Pueden reducir ruido (denoising).
- Se aplican a imágenes, audio y datos tabulares.

Limitaciones

- Requieren muchos datos.
- Entrenamiento costoso.
- Riesgo de sobreajuste.

Figura 5

Diagrama de flujo del funcionamiento de un Autoencoder.



3. Comparativa y conclusiones

Tabla 1

Comparación general de técnicas de clustering y reducción de dimensionalidad.

Técnica	Tipo	Requiere etiquetas	Ventajas	Limitaciones
K-means	Clustering	No	Rápido, escalable	Sensible a outliers
Jerárquico	Clustering	No	Dendrograma interpretable	Costoso
DBSCAN	Clustering	No	Maneja ruido, formas complejas	Difícil calibración
LDA	Reducción	Sí	Separación discriminante	Supone linealidad
Autoencoders	Reducción	No obligatorio	Representaciones no lineales	Alto costo computacional

Cuando usar cada técnica

- **Clustering:** cuando se desea segmentar datos no etiquetados, encontrar patrones o detectar anomalías.
- **Reducción de dimensionalidad:** cuando los datos tienen muchas variables, contienen ruido o se requiere visualización o compresión.

Conclusiones

El clustering y la reducción de dimensionalidad son técnicas complementarias que permiten entender mejor los datos desde distintas perspectivas. Mientras el clustering ayuda a identificar patrones, grupos naturales y estructuras ocultas, la reducción de dimensionalidad facilita simplificar la información, eliminar ruido y mejorar la visualización. La elección entre una u otra depende del objetivo: si se busca segmentar o descubrir relaciones internas, se utiliza clustering; si se desea reducir complejidad o preparar los datos para otros modelos, se aplica reducción de dimensionalidad. Cada algoritmo aporta ventajas particulares que lo hacen adecuado para situaciones específicas dentro de la ciencia de datos.

Referencias

Neptune.ai. (2023). *K-means clustering explained*.
https://neptune.ai/blog/k-means-clustering

GeeksforGeeks. (2024). *Hierarchical clustering*.
https://www.geeksforgeeks.org/machine-learning/hierarchical-clustering/

DataCamp. (2024). *DBSCAN clustering algorithm guide*.
https://www.datacamp.com/tutorial/dbscan-clustering-algorithm

Raschka, S. (2014). *Linear Discriminant Analysis (LDA) explained in Python*.
https://sebastianraschka.com/Articles/2014_python_lda.html