# UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

# DESARROLLO Y GESTIÓN DE SOFTWARE

# Extracción de Conocimiento en Bases de Datos

## IV.1. Algoritmos de agrupación


**Docente:**

Enrique Mascote

**Presentan:**

Ian Carlos Chávez Rojo

**Grupo:**

IDGS91N


Fecha: 28/11/2025

# Índice

Introducción ..... 3

Algoritmos de agrupación ..... 3

K-means ..... 3

Clustering jerárquico (aglomerativo o divisivo) ..... 4

DBSCAN ..... 5

Algoritmos de reducción de dimensionalidad ..... 6

Análisis de Componentes Principales (PCA) ..... 6

Autoencoders ..... 7

Comparativa y conclusiones ..... 8

Situaciones prácticas en las que uno tarda prioridad sobre otro ..... 8

Referencias bibliográficas ..... 9

# Introducción

En la extracción de conocimiento a partir de grandes volúmenes de datos, el clustering y la reducción de dimensionalidad son pilares importantes. El clustering permite descubrir patrones ocultos agrupando datos similares sin necesidad de etiquetas previas, lo que facilita la identificación de perfiles, tendencias y comportamientos. Por otro lado, la reducción de dimensionalidad mejora la eficiencia del análisis al simplificar los datos sin perder la esencia de la información, haciendo más accesible la visualización e interpretación de conjuntos complejos. Ambos enfoques complementan la exploración de datos, aportando estructura y claridad al proceso de descubrimiento de conocimiento.

# Algoritmos de agrupación

Los algoritmos de agrupación o clustering permiten clasificar datos no etiquetados en grupos homogéneos según su similitud, descubriendo patrones ocultos. Son útiles en análisis de mercados, detección de anomalías o segmentación de usuarios. Métodos populares como K-means, DBSCAN y clustering jerárquico utilizan distintos criterios para definir similitud y formar clústers de forma automatizada.

# K-means

**Principio de funcionamiento:** K-means es un algoritmo no supervisado que agrupa datos en k clústers según su similitud. Inicia con la selección de k centroides, asignando cada dato al más cercano. Luego, recalcula los centroides como el promedio de sus miembros y repite el proceso hasta estabilizar las posiciones o alcanzar un límite de iteraciones.

**Parámetros clave:**

Entre los parámetros clave de K-means se encuentra k, el número de clústers a formar, cuya elección es crucial para obtener agrupamientos significativos. La métrica más común para medir la cercanía entre puntos es la distancia euclidiana. Los centroides iniciales pueden seleccionarse aleatoriamente o mediante métodos como k-means++, y su adecuada elección influye en la calidad del resultado. Por último, el criterio de convergencia se establece según el cambio mínimo en los centroides o un número máximo de iteraciones.

**Ventajas y limitaciones**: K-means es eficiente y fácil de implementar, especialmente útil cuando se trabaja con grandes volúmenes de datos numéricos. Sin embargo, su

rendimiento depende fuertemente de la elección de k y los centroides iniciales. No funciona bien con datos de formas arbitrarias o de densidades variables, y puede verse afectado por valores atípicos. Además, todos los clústers deben tener una estructura similar para que la segmentación sea eficaz, lo cual limita su aplicación en entornos más heterogéneos.

## Ejemplo de aplicación simple

***Entrada**: Conjunto de datos D = {(1,2), (2,1), (8,9), (9,8)}*
***Número de clústers**: k = 2*

***Inicializar centroides**:*
*C1 = (1,2)*
*C2 = (8,9)*

***Mientras los centroides cambien significativamente**:*
*Para cada punto en D:*
*Calcular distancia a C1 y C2*
*Asignar el punto al clúster con el centro más cercano*

*Para cada clúster:*
*Recalcular el centroide como el promedio de sus puntos*

***Fin***
***Salida**:*
*Clúster 1: {(1,2), (2,1)} con centroide cercano a (1.5, 1.5)*
*Clúster 2: {(8,9), (9,8)} con centroide cercano a (8.5, 8.5)*

## Clustering jerárquico (aglomerativo o divisivo)

**Principio de funcionamiento**: El clustering jerárquico crea una estructura en forma de árbol (dendrograma) que representa similitudes entre datos. En su variante aglomerativa, los puntos se fusionan progresivamente en grupos; en la divisiva, se dividen desde un único conjunto inicial. Las decisiones de agrupamiento se basan en distancias entre elementos, y el resultado permite explorar diferentes niveles de segmentación de forma visual.

**Parámetros clave:**

Entre los parámetros esenciales del clustering jerárquico se encuentra el tipo de vínculo (como promedio, completo, simple o Ward), que define cómo se mide la distancia entre clústers. La métrica de distancia, habitualmente euclidiana, también puede variar (Manhattan, Coseno, etc.), afectando el resultado de la agrupación. El nivel de corte del dendrograma permite establecer cuántos grupos finales se obtienen según el umbral de

similitud deseado. Finalmente, el tipo de algoritmo empleado (aglomerativo o divisivo) determina el enfoque para construir la jerarquía de agrupación.

**Ventajas y limitaciones**: Una ventaja del clustering jerárquico es que no requiere definir el número de clústers previamente, ya que su estructura tipo árbol permite explorar múltiples niveles de agrupación mediante un dendrograma. Aunque ofrece una visualización clara, su complejidad computacional es elevada, lo que limita su uso en grandes volúmenes de datos. Además, es sensible a valores atípicos y a la configuración de la métrica y método de enlace.

### Ejemplo de aplicación simple

***Entrada**: Conjunto de datos D = {(1,2), (2,2), (8,8), (9,9)}*
***Inicializar**:*
*Cada punto es su propio clúster*

***Repetir**:*
*Calcular distancia entre todos los clústers*
*Seleccionar los dos clústers más cercanos*
*Fusionarlos en uno solo*

***Hasta que**:*
*Se forme una única jerarquía o se alcance un número deseado de clústers*

***Salida**:*
*Dendrograma con niveles de agrupación:*
*- Nivel 1: {(1,2), (2,2)} y {(8,8), (9,9)}*
*- Nivel final: todos los puntos en un solo clúster*

### DBSCAN

**Principio de funcionamiento**: DBSCAN es un algoritmo basado en densidad que agrupa puntos cercanos entre sí y separa los que están en regiones dispersas, tratándolos como ruido. A diferencia de otros métodos, no necesita definir el número de clústers previamente; identifica núcleos según el número de vecinos dentro de un radio específico y expande los grupos desde ellos, lo que lo hace eficaz en distribuciones irregulares y robusto ante anomalías.

**Parámetros clave:**

Entre los parámetros clave de DBSCAN se encuentra ε (epsilon), que define el radio dentro del cual se buscan vecinos para cada punto. Si el número de vecinos en esa zona alcanza el MinPts especificado, el punto se considera núcleo y puede formar parte de un clúster. La

métrica de distancia utilizada (como Euclidiana, Manhattan o Coseno) influye en la detección de densidades, permitiendo adaptar el algoritmo a distintos tipos de datos y estructuras.

**Ventajas y limitaciones**: Entre sus fortalezas, DBSCAN destaca por ser capaz de identificar clústers de formas arbitrarias y detectar ruido sin necesidad de conocer a priori la cantidad de grupos. Es especialmente útil en conjuntos donde hay variabilidad en la densidad. No obstante, su desempeño depende mucho de la correcta elección de ε y MinPts; parámetros mal calibrados pueden provocar agrupaciones deficientes o ignorar subestructuras relevantes. Además, se vuelve menos eficiente con datos de alta dimensión, donde el concepto de "densidad" puede volverse ambiguo.

## Ejemplo de aplicación simple

***Entrada**:*
*Conjunto de datos D = {(1,2), (2,2), (2,1), (8,8), (9,9)}*
*Parámetros: ε = 1.5, MinPts = 2*

***Para cada punto P en D:***
*Encontrar vecinos dentro de radio ε*
*Si número de vecinos ≥ MinPts:*
*P es un punto núcleo → expandir clúster desde P*
*Si número de vecinos < MinPts:*
*P es ruido (temporalmente)*

***Repetir hasta que todos los puntos estén asignados**:*
*Expandir clústers conectando núcleos y sus vecinos*

***Salida**:*
*Clúster 1: {(1,2), (2,2), (2,1)}*
*Clúster 2: {(8,8), (9,9)}*
*Ruido: Ninguno en este caso*

## Algoritmos de reducción de dimensionalidad

### Análisis de Componentes Principales (PCA)

**Fundamento conceptual**: PCA es una técnica estadística que transforma datos con muchas variables en un conjunto reducido de componentes principales, manteniendo la mayor variabilidad posible. Se basa en la descomposición de la matriz de covarianza para encontrar direcciones (vectores propios) que representan mejor la estructura de los datos.

**Parámetros clave**: Los aspectos más importantes incluyen el número de componentes principales a conservar, el tipo de normalización previa (como centrado en la media), y el porcentaje de varianza acumulada deseado. Estos definen cuánta información se retiene tras la reducción.

**Ventajas y limitaciones**: Entre sus ventajas destacan la mejora del rendimiento computacional, la eliminación de redundancias entre variables y la facilidad para visualizar datos en 2D o 3D. Sin embargo, puede resultar difícil interpretar los componentes obtenidos, y no siempre es eficaz con relaciones no lineales entre datos.

**Ejemplo sencillo en pseudocódigo**

***Entrada**: Matriz X (n muestras × m variables)*

*1. Normalizar X (centrar en la media)*
*2. Calcular matriz de covarianza de X*
*3. Obtener vectores propios (eigenvectors) y valores propios (eigenvalues)*
*4. Ordenar componentes por mayor varianza*
*5. Seleccionar k componentes principales*
*6. Transformar X proyectándola sobre esos componentes*

***Salida**: Datos reducidos con k dimensiones*

**Autoencoders**

**Fundamento conceptual**: Los autoencoders son redes neuronales entrenadas para reconstruir sus propias entradas. Aprenden una representación comprimida al codificar los datos en una capa intermedia (codificador) y luego los reconstruyen en la salida (decodificador). Este proceso permite capturar las características más relevantes, convirtiéndose en una herramienta eficaz para reducción de dimensionalidad en entornos no lineales.

**Parámetros clave**: Incluyen la arquitectura de la red (número de capas y neuronas), la función de activación (ReLU, Sigmoid, etc.), el tamaño del bottleneck (dimensión reducida), y el criterio de pérdida (como el error cuadrático medio) que mide qué tan bien se reconstruyen los datos originales.

**Ventajas y limitaciones**: Su ventaja principal es la capacidad para capturar relaciones no lineales y reconstruir datos con gran precisión, incluso en contextos complejos como

imágenes. Sin embargo, requieren mayor capacidad computacional, conocimiento en redes neuronales, y son sensibles al sobreajuste si no se regularizan adecuadamente.

**Ejemplo sencillo en seudocódigo**

***Entrada**: Conjunto X de datos normalizados*

*1. Definir red neuronal con capa de codificación (dimensión reducida)*
*2. Agregar capa de decodificación para reconstruir X*
*3. Entrenar red minimizando la diferencia entre X y su reconstrucción*
*4. Extraer la salida de la capa intermedia como datos reducidos*

***Salida**: Representación comprimida de X*

### Comparativa y conclusiones

Cuando se desea identificar patrones en datos sin etiquetas, el clustering permite agrupar elementos similares para facilitar su interpretación. Si el conjunto posee muchas variables difíciles de analizar, la reducción de dimensionalidad simplifica el espacio manteniendo la información clave. En la práctica, ambos se combinan: primero se reduce la dimensionalidad para mejorar el análisis, luego se aplica clustering para segmentar con mayor precisión.

### Situaciones prácticas en las que uno tarda prioridad sobre otro

Cuando se busca comprender la estructura oculta de los datos sin etiquetas, el clustering es la opción preferida, ya que permite formar grupos automáticamente, como en el análisis de clientes. En cambio, si el problema involucra muchas variables, como en imágenes médicas o sensores IoT, la reducción de dimensionalidad es prioritaria para eliminar ruido y facilitar el análisis. En la práctica, suelen combinarse: primero se reducen las dimensiones para simplificar, y luego se aplican técnicas de agrupamiento para detectar patrones significativos.

## Conclusiones generales

En minería de datos no supervisada, el clustering y la reducción de dimensionalidad son herramientas complementarias con funciones distintas: el primero identifica patrones y agrupa datos similares, mientras que el segundo simplifica variables para mejorar el procesamiento y la visualización. Al combinarlos, se facilita el análisis en contextos complejos como diagnósticos médicos, análisis de mercados o sistemas de recomendación. Su aplicación depende de los objetivos, tipo de datos y recursos disponibles.

## Referencias bibliográficas


- Barrios, A. (2023, marzo 10). Tutorial del Algoritmo DBSCAN en Python - LatinXinAI - Medium. LatinXinAI. https://medium.com/latinxinai/tutorial-del-algoritmo-dbscan-en-python-d7c187c1b24f
- Benhur, S. (2023, febrero 24). Hierarchical clustering: Agglomerative and divisive explained. Built In. https://builtin.com/machine-learning/agglomerative-clustering
- Bergmann, D., & Stryker, C. (2025, mayo 30). ¿Qué es un autocodificador? Ibm.com. https://www.ibm.com/mx-es/think/topics/autoencoder
- Clustering con Python. (s/f). Cienciadedatos.net. Recuperado el 14 de julio de 2025, de https://cienciadedatos.net/documentos/py20-clustering-con-python
- Daniel. (2022, enero 6). Comprende el algoritmo t-SNE en 3 pasos. DataScientest. https://datascientest.com/es/comprende-el-algoritmo-t-sne-en-3-pasos
- Ejemplo con DBSCAN. (s/f). Interactivechaos.com. Recuperado el 14 de julio de 2025, de https://interactivechaos.com/es/manual/tutorial-de-machine-learning/ejemplo-con-dbscan
- Ejemplo con t-SNE. (s/f). Interactivechaos.com. Recuperado el 14 de julio de 2025, de https://interactivechaos.com/es/manual/tutorial-de-machine-learning/ejemplo-con-t-sne
- GraphEverywhere, E. (2020, agosto 21). ¿Qué es el Clustering? GraphEverywhere; Graph Everywhere SL. https://www.grapheverywhere.com/que-es-el-clustering/
- Interactive Chaos. (s/f). Interactivechaos.com. Recuperado el 14 de julio de 2025, de https://interactivechaos.com/es/manual/tutorial-de-machine-learning/t-sne
- Kavlakoglu, E., & Winland, V. (2025, febrero 13). ¿Qué es la agrupación en clústeres k-means? Ibm.com. https://www.ibm.com/mx-es/think/topics/k-means-clustering

- López, M. (2023, junio 23). Qué es el Clustering y Cómo Comprender los Datos - IMMUNE. IMMUNE Technology Institute. https://immune.institute/blog/que-es-el-clustering/
- Machine Learning & Clustering: el algoritmo DBSCAN. (2022, noviembre 30). DataScientest. https://datascientest.com/es/machine-learning-clustering-dbscan
- Murel, J., & Kavlakoglu, E. (2025, febrero 18). ¿Qué es la reducción de la dimensionalidad? Ibm.com. https://www.ibm.com/mx-es/think/topics/dimensionality-reduction
- Na. (2018, marzo 12). K-Means con Python paso a paso. Aprendemachinelearning.com; Juan Ignacio Bagnato. https://www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/
- Noble, J. (2025, febrero 13). ¿Qué es la agrupación jerárquica? Ibm.com. https://www.ibm.com/mx-es/think/topics/hierarchical-clustering
- ¿Qué es la agrupación en clústeres? (2025, febrero 18). Ibm.com. https://www.ibm.com/mx-es/think/topics/clustering
- Reducción de dimensionalidad. (s/f). Interactivechaos.com. Recuperado el 14 de julio de 2025, de https://interactivechaos.com/es/wiki/reduccion-de-dimensionalidad
- Sanz, F. (2020, noviembre 26). Algoritmo K-Means Clustering – aplicaciones y desventajas. The Machine Learners. https://www.themachinelearners.com/k-means/