UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la información

Extracción de Conocimiento en Bases de Datos

IV.1. Algoritmos de agrupación (25%)

IDGS91N

Presenta:

Carlos Isaac Parra Aguirre

Docente:

Enrique Mascote

30 de November de 2025

Tabla de contenido

Introducción

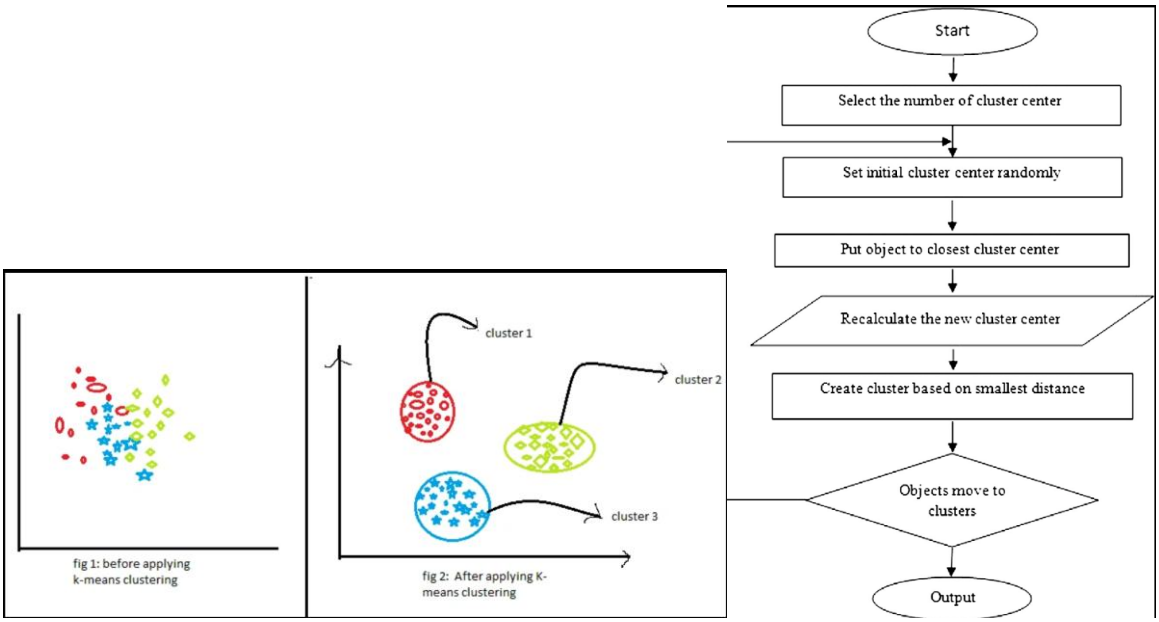
En la extracción de conocimiento, los analistas se enfrentan frecuentemente a grandes volúmenes de datos que contienen patrones difíciles de identificar a simple vista. Para resolver este problema, se utilizan dos enfoques fundamentales: **el clustering** y **la reducción de dimensionalidad**.
El **clustering** permite descubrir grupos naturales dentro de los datos sin necesidad de etiquetas previas, mientras que la **reducción de dimensionalidad** ayuda a simplificar datasets complejos eliminando redundancia, ruido y variables poco relevantes.

Ambas técnicas forman parte esencial del aprendizaje no supervisado y se utilizan en minería de datos, visión por computadora, marketing, medicina, sistemas de recomendación y más. En este reporte se describen los algoritmos más representativos de ambos enfoques, junto con ejemplos visuales y comparativas que facilitan su comprensión.

1. Algoritmos de Agrupación (Clustering)

A continuación, se presentan **tres algoritmos de clustering**: *K-means*, *Clustering jerárquico aglomerativo*, y *DBSCAN*.

1.1 K-means



Principio de funcionamiento

K-means divide los datos en *k* grupos basándose en la distancia entre puntos y los centroides.
El proceso es iterativo:

1. Se eligen aleatoriamente *k* centroides.
2. Cada punto se asigna al centroide más cercano.
3. Se recalculan los centroides.
4. Se repite hasta convergencia.

Parámetros clave

- **k:** número de clústeres.
- **Distancia:** generalmente euclidiana.
- **Iteraciones máximas**.

Ventajas

- Muy rápido y escalable.
- Fácil de implementar.
- Útil para datos grandes.

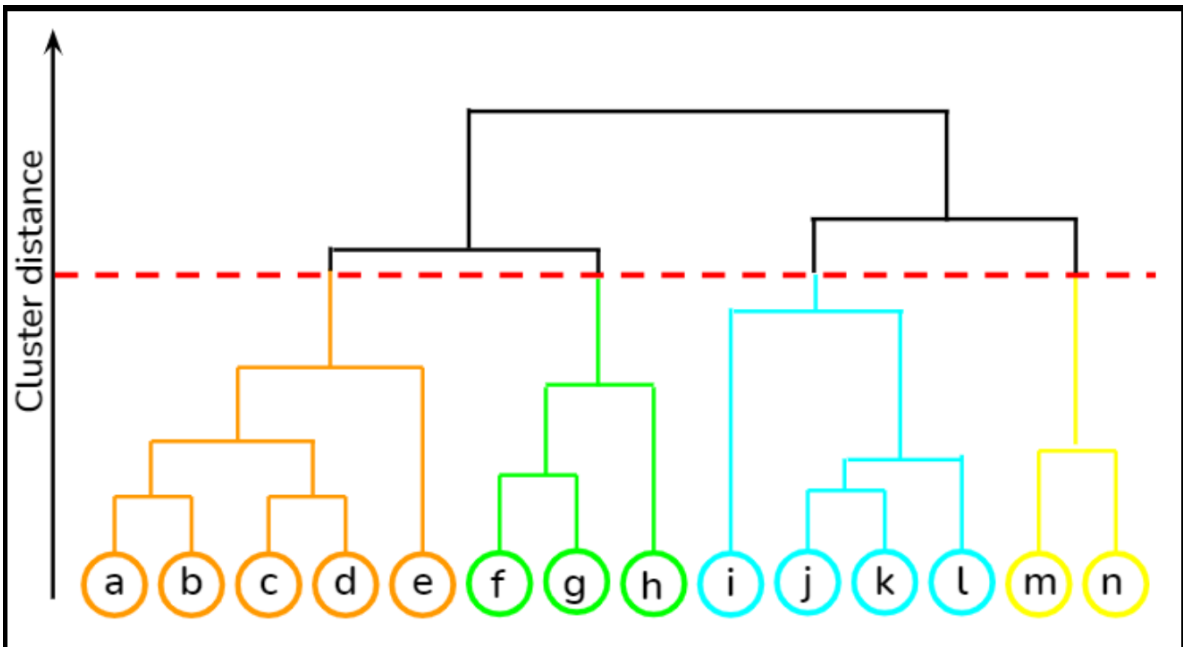
Limitaciones

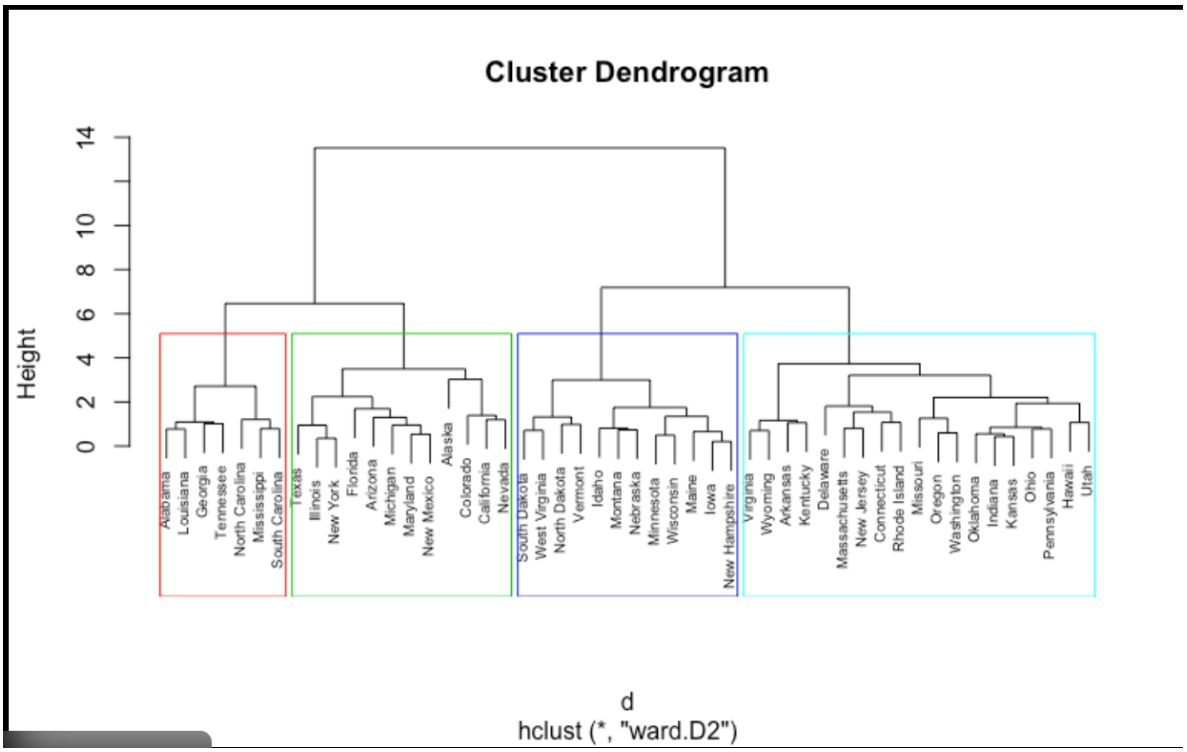
- Requiere definir *k* de antemano.
- Sensible a outliers.
- No funciona bien con clústeres no esféricos.

Ejemplo simple (pseudocódigo)

```
Inicializar k centroides al azar
REPETIR:
    Asignar cada punto al centroide más cercano
    Recalcular centroides
HASTA que los centroides ya no cambien
```

1.2 Clustering jerárquico aglomerativo (HAC)





Principio de funcionamiento

HAC inicia con cada punto como un clúster independiente y fusiona los más similares hasta formar uno solo.
Su producto visual es el **dendrograma**, un árbol que muestra la relación jerárquica entre los grupos.

Parámetros clave

- **Métrica de distancia:** euclidiana, Manhattan, etc.
- **Método de enlace:**
  - *single linkage* (mínima distancia),
  - *complete linkage* (máxima distancia),
  - *average linkage*.

Ventajas

- No requiere elegir *k* inicialmente.
- Produce una estructura jerárquica muy informativa.

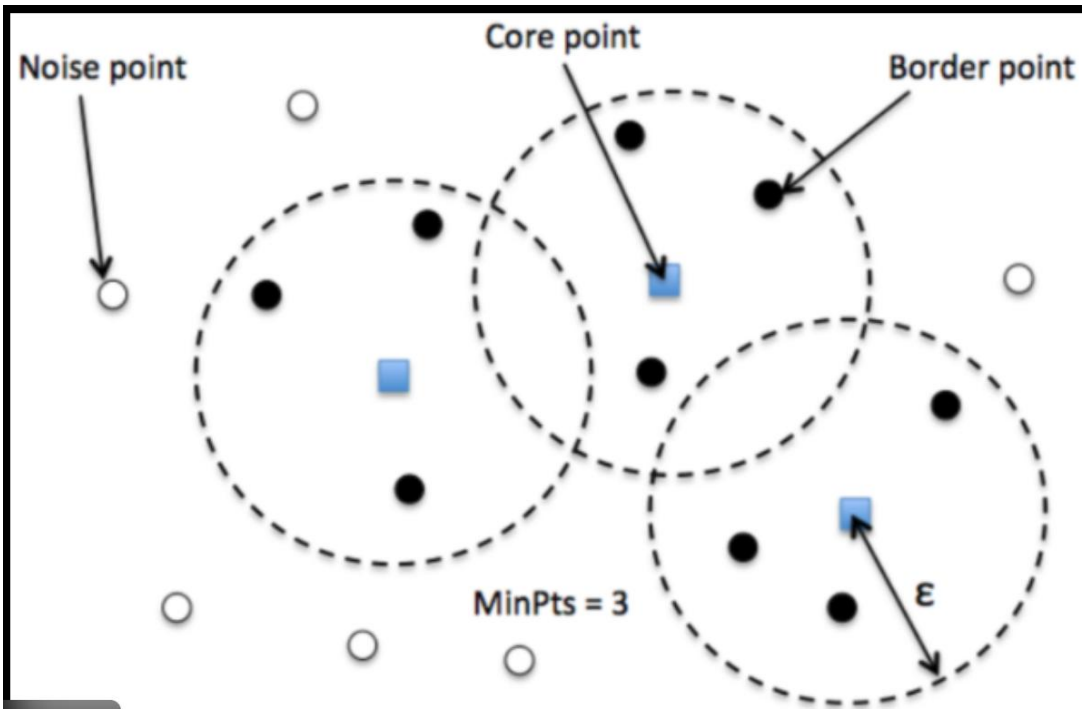
Limitaciones

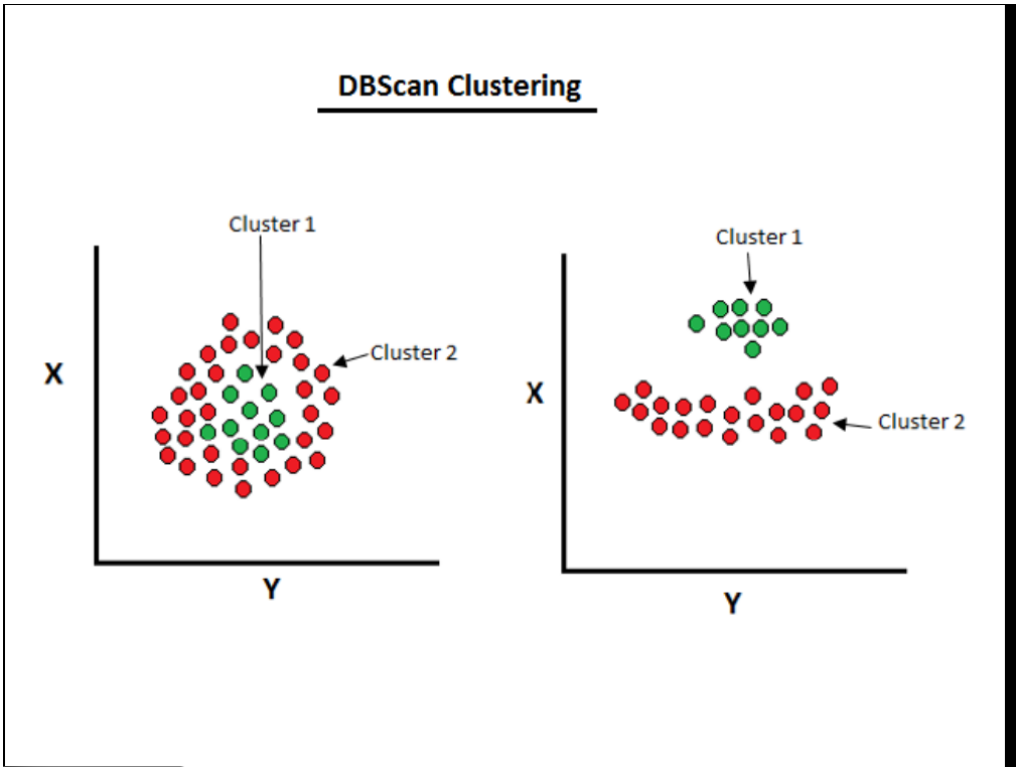
- No escala bien a datasets grandes.
- Sensible al ruido.

Ejemplo

Un dataset de clientes puede mostrar cómo grupos pequeños se fusionan hasta formar segmentos de mercado más amplios.

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





Principio de funcionamiento

DBSCAN agrupa puntos que están densamente conectados entre sí. Define tres tipos de puntos:

- **Core:** densidad alta.
- **Border:** cerca de un core.
- **Noise:** aislados o outliers.

Parámetros clave

- **eps:** radio máximo para considerar vecinos.
- **minPts:** número mínimo de puntos dentro de *eps*.

Ventajas

- Detecta clústeres de formas arbitrarias.
- Maneja outliers de forma natural.
- No requiere *k*.

Limitaciones

- Difícil elegir eps y minPts.
- No funciona bien con densidades muy diferentes.

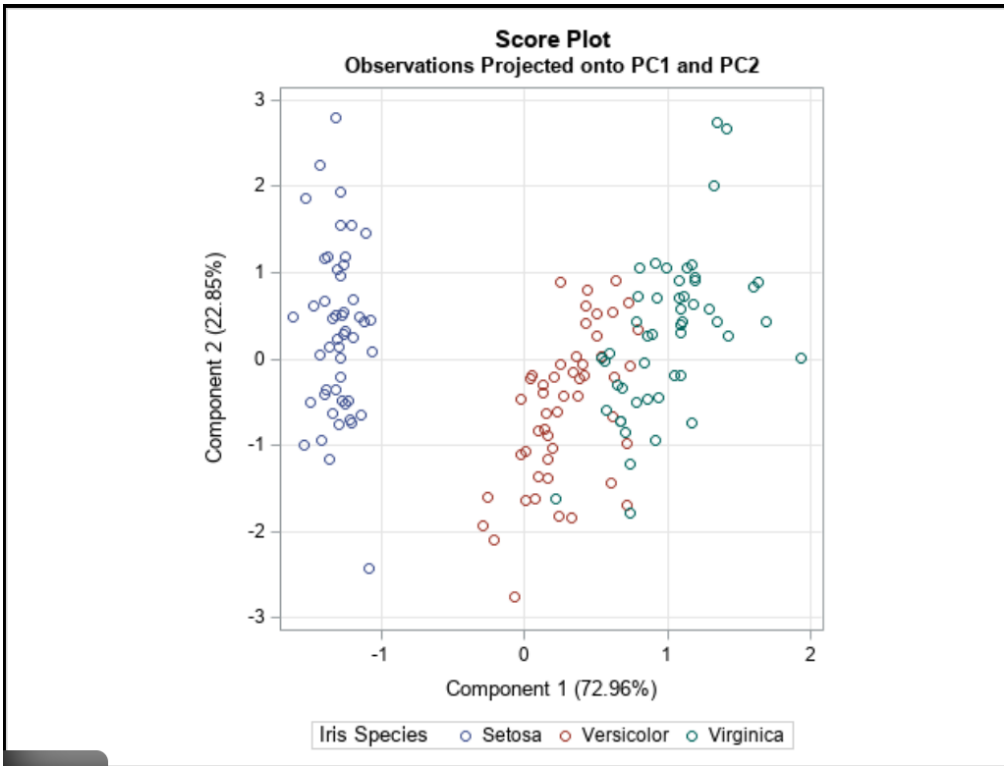
Ejemplo

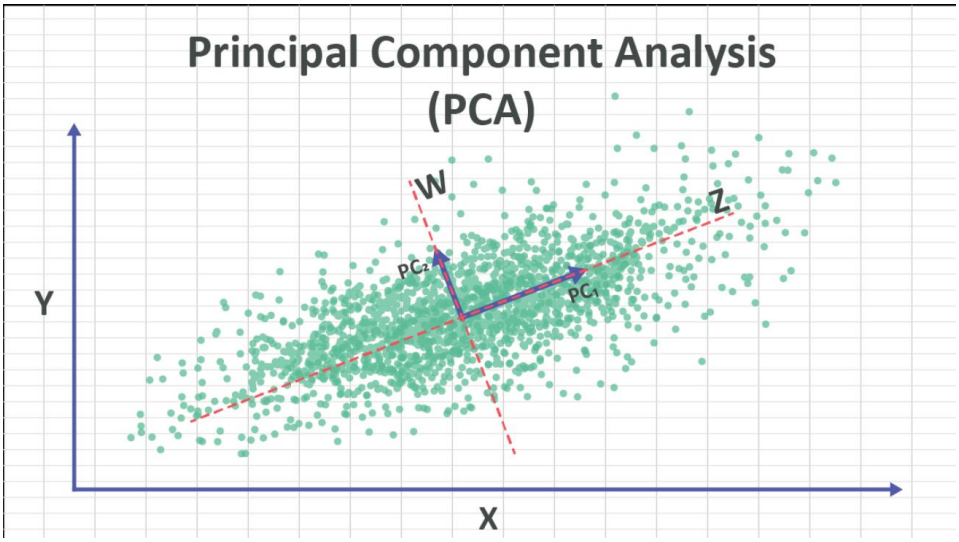
Ideal para mapas de calor geoespaciales o detección de zonas anómalas.

2. Algoritmos de Reducción de Dimensionalidad

A continuación, se presentan **dos algoritmos**: *PCA* y *t-SNE*.

2.1 PCA (Análisis de Componentes Principales)





Fundamento matemático

PCA transforma las variables originales en nuevas variables llamadas **componentes principales**, que capturan la mayor varianza posible. Se basa en:

- Matrices de covarianza
- Descomposición en valores propios (autovalores y autovectores)

Parámetros clave

- Número de componentes a conservar (*n_components*).

Ventajas

- Reduce ruido y redundancia.
- Simplifica datos para visualización en 2D o 3D.
- Muy rápido.

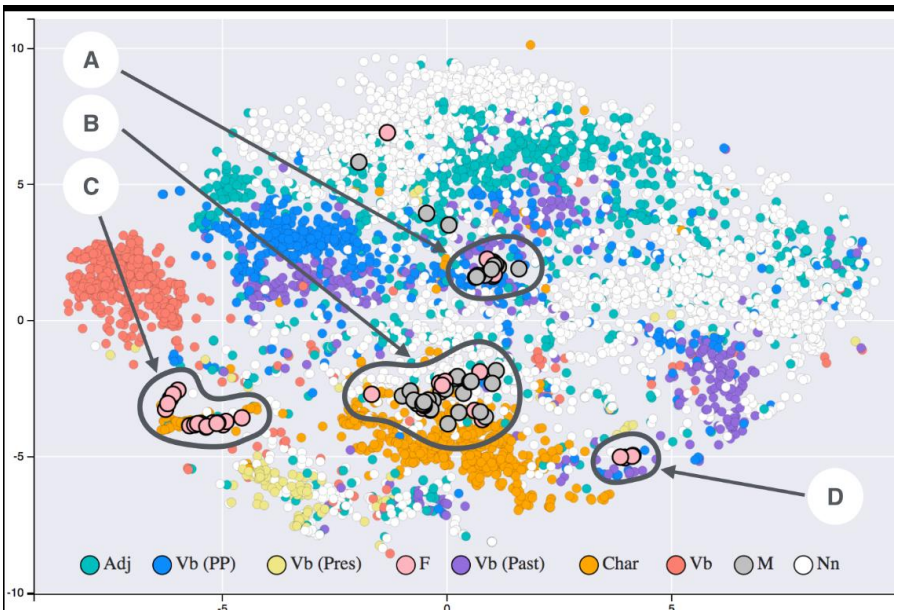
Limitaciones

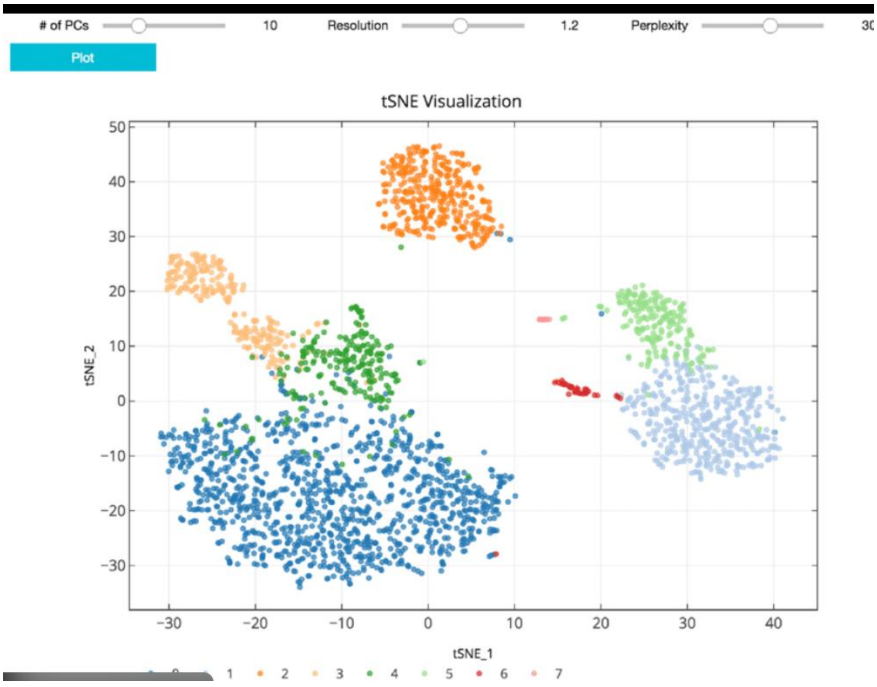
- Solo captura relaciones lineales.
- La interpretación de componentes puede ser difícil.

Ejemplo

Reducir un dataset de 20 variables a 2 para visualización bidimensional.

2.2 t-SNE (t-Distributed Stochastic Neighbor Embedding)





}

Fundamento conceptual

t-SNE reduce dimensionalidad preservando la **vecindad local**, es decir, puntos que eran similares permanecen cercanos en el mapa final.
Es muy usado para:

- Visualizar embeddings
- Agrupamiento natural de datos complejos

Parámetros clave

- **perplexity:** controla el tamaño del vecindario.
- **learning rate**.
- **iterations**.

Ventajas

- Visualizaciones muy claras de clústeres naturales.
- Ideal para datos no lineales.

Limitaciones

- Computacionalmente costoso.
- No es adecuado para modelado directo, solo para visualización.

Ejemplo

Visualizar la distribución de dígitos escritos a mano (MNIST).

3. Comparativa y Conclusiones

Cuándo usar clustering vs reducción de dimensionalidad

Técnica	Objetivo principal	¿Cuándo usarla?
Clustering	Encontrar grupos naturales	Segmentación, detección de patrones, agrupación sin etiquetas
Reducción de dimensionalidad	Simplificar datos y extraer características	Visualización, eliminación de ruido, preprocesamiento

Relación entre ambas

En muchos flujos reales de minería de datos: **Primero se reduce la dimensionalidad (PCA/t-SNE)** para eliminar ruido **Luego se aplica clustering (K-means/DBSCAN)** para mejorar los clústeres.

Conclusiones generales

- El clustering es ideal cuando se busca descubrir patrones ocultos o grupos en datos no etiquetados.
- La reducción de dimensionalidad ayuda a mejorar la eficiencia, interpretación y visualización, especialmente en datasets grandes.
- Ambos métodos son complementarios y forman parte esencial del aprendizaje no supervisado.

Referencias

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

1. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

2. McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection*. arXiv.
3. Pedregosa, F. et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research.
4. Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). *A Density-Based Algorithm for Discovering Clusters*. KDD.
5. Van der Maaten, L., & Hinton, G. (2008). *Visualizing Data using t-SNE*. Journal of Machine Learning Research.
6. Jolliffe, I. (2002). *Principal Component Analysis*. Springer.