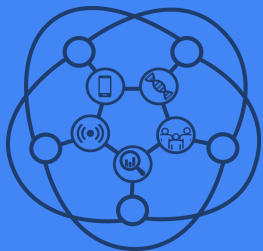
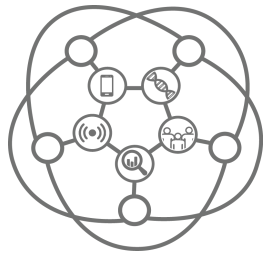




Aprendizaje No Supervisado

Laboratorio de Redes y Ciencia de Datos
Junio 2019





El Equipo



Emanuel Becerra
elbecerrasoto@gmail.com



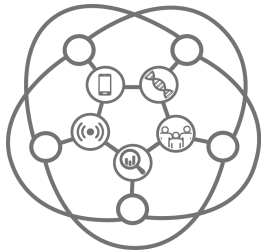
Omar Gutiérrez
omargup@gmail.com



Andrea Torres
andys0tc@gmail.com



Edgar Romo
omargup@gmail.com



El Equipo



Bruno Gutiérrez

bruno.gupa@gmail.com



Karina Torres

karina.torres.calderon@gmail.com



Carlos Assad

carlosloart@gmail.com



Ricardo Menchaca

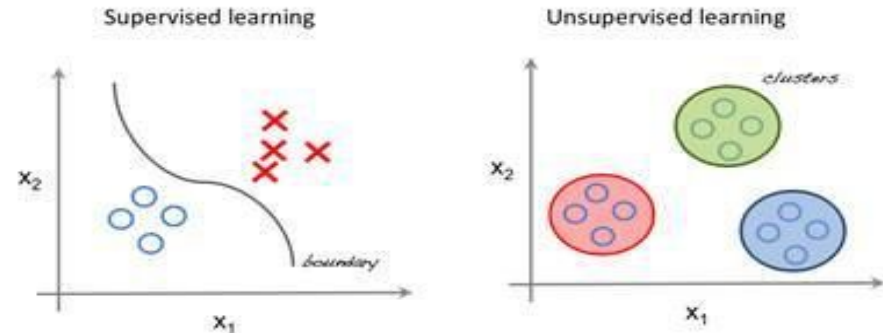
ric@cic.ipn.mx

Introducción

Técnicas de aprendizaje máquina que se utilizan sin hacer referencia a ninguna etiqueta en específico.

Se encarga de encontrar estructura en los datos.

Los términos Supervisado y No Supervisado no son términos formales y la línea de separación puede ser borrosa.



Tipos de Aprendizaje No Supervisado

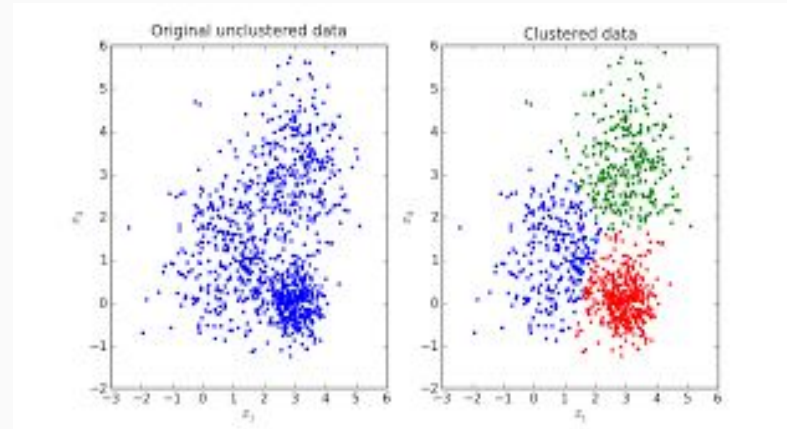
- ❖ Agrupamiento
- ❖ Reducción Dimensional
- ❖ Detección de anomalías

Agrupamiento

Técnicas para particionar el conjunto de datos en subconjuntos, con elementos cercanos entre sí y lejanos de otros subconjuntos.

Los grupos obtenidos vistos como clases, por ello se dice clasificación automática.

Valores atípicos, aquellos que se encuentre muy lejanos de ciertos grupos principales.



Agrupamiento Ejemplos

Business Intelligence Organizar un gran número de clientes en grupos de clientes, donde los grupos comparten características similares y de interés para *marketing*

Una compañía consultora puede particionar sus proyectos en categorías similares para mejorar su finalización.

Agrupamiento Ejemplos

Un sistema de recomendaciones (Netflix), agrupando clientes con patrones similares de gustos de entretenimiento.

En reconocimiento de imágenes puede ser usado para descubrir clases en sistemas de reconocimiento de caracteres.

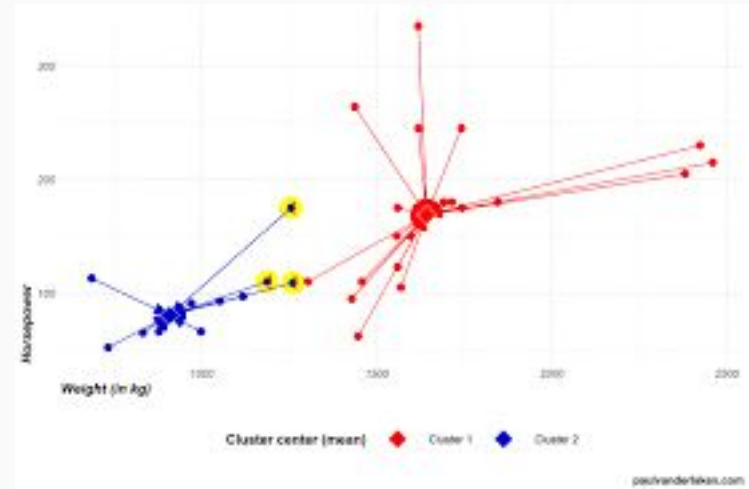
Agrupamiento Ejemplos

Búsqueda Web para clasificar los hits de una palabra y evitar regresar al usuario resultados muy parecidos.

Detección de valores atípicos, para monitoreo de transacciones financieras.

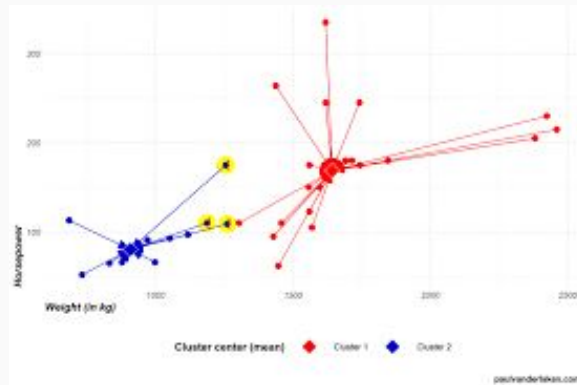
K-means

- Algoritmo simple de clustering
- Converge en un número finito de pasos
- Parte solo de los datos
- Funcionamiento



Algoritmo

1. Seleccionamos el valor de K
2. Asignamos K centroides aleatoriamente en los datos
3. Actualizamos la posición de los centroides dada la distancia a los datos
4. Si ya no cambia la posición de los centroides terminamos.

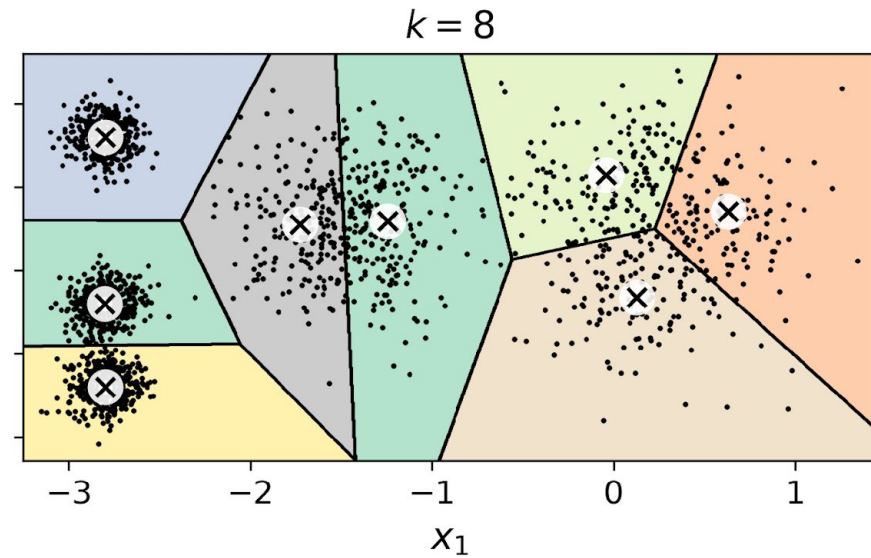
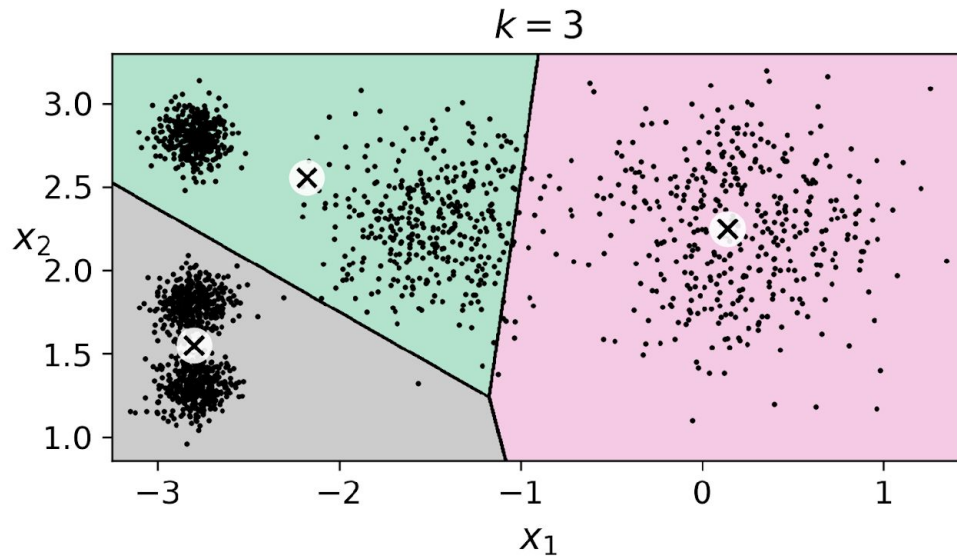


K-means

- Inicialización afecta el tiempo de convergencia del algoritmo
- Inicialización random
- K-means ++ inicializa con centroides alejados entre ellos
- Uso de mini-Batches, acelera el tiempo de convergencia

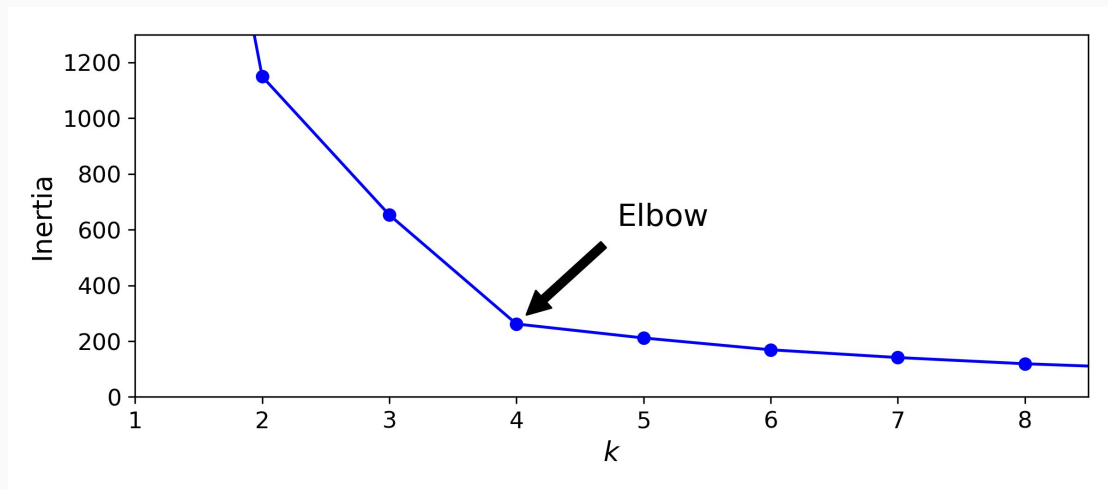
K- means

¿Qué valor debe tomar k ?



Medidas de desempeño

Inercia: distancia media de las instancias a su centroide

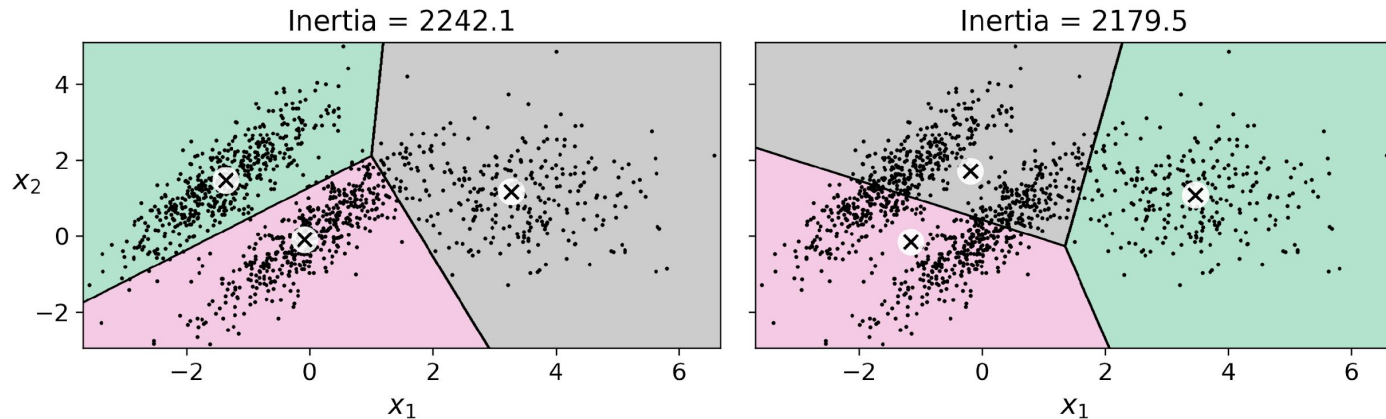


Coeficiente de silueta: considera la distancia media a las instancias dentro cluster (a) y la distancia media a las instancias del cluster más cercano (b)

$$(b - a) / \max(a, b)$$

Limites de K-means

- Tiempos largos para converger a una solución óptima
- Especificar el valor de k
- Cambio en los clusters altera el algoritmo
- Figuras anormales en los clusters



Segmentación de imágenes con K-means

Los píxeles que son parte del mismo objeto, pertenecen al mismo segmento.

Actualmente la investigación se orienta a redes convolucionales.

Un enfoque más simple es con segmentación de color.

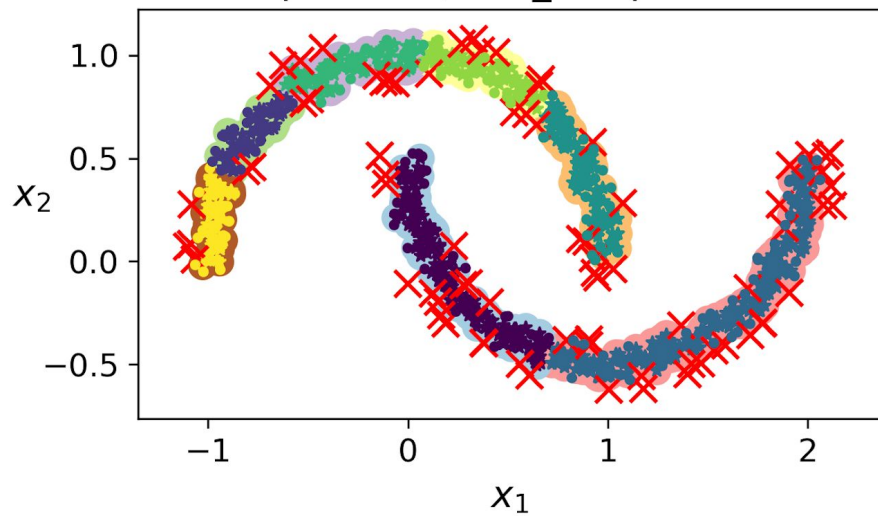
Requiere un vector de tres dimensiones
altura, anchura, canal de color RGB



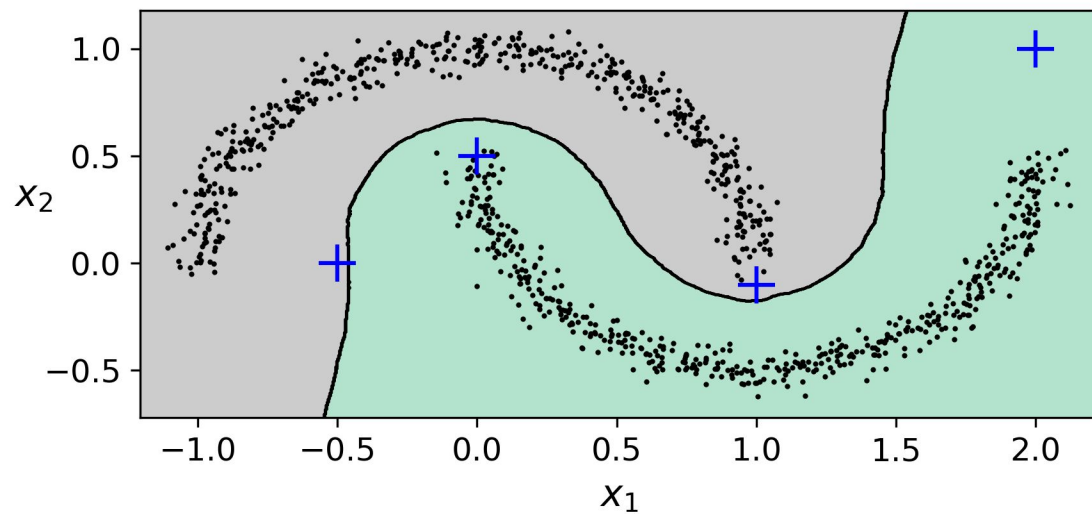
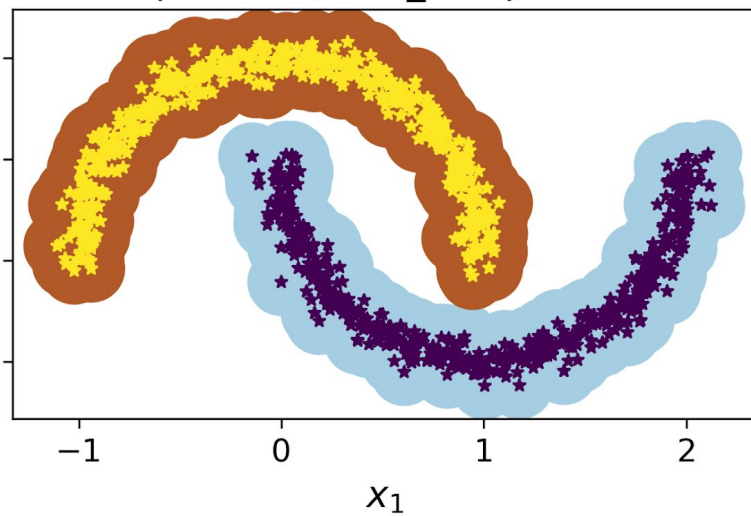
DBSCAN

- **Density-based spatial clustering of applications with noise**
 - Contabiliza el número de instancias dentro de una distancia ϵ
 - Si la instancia tiene al menos $min_samples$ en ϵ , es una *core instance*
 - Todas las instancias en el vecindario de un *core instance* pertenecen al mismo cluster incluyendo múltiples *core instances*
 - Anomalías son *core instances* aisladas

eps=0.05, min_samples=5

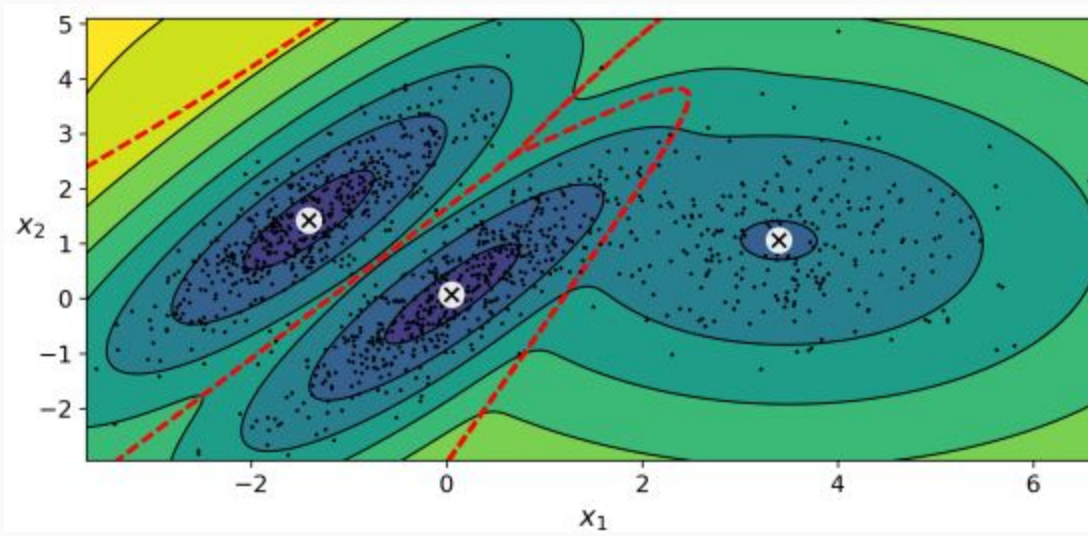


eps=0.20, min_samples=5



Gaussian Mixtures

Es un modelo probabilístico que asume que los datos fueron generados a partir de varias distribuciones Gaussianas y trata de encontrar los parámetros de dichas Gaussianas.



Reducción Dimensional

- Preprocesamiento antes de una tarea supervisada
- Como filtro para ruido
- Para combatir el problema (maldición) de las dimensiones
- Visualización
- EDA (Exploratory Data Analysis)
- Compresión de los datos

La maldición de las dimensiones

- Puntos al azar en una línea, distancia ~ 0.33
- Puntos al azar en una cuadrado, distancia ~ 0.52
- Puntos al azar en un cubo ~ 0.66
-
-
- Puntos al azar en un hipercubo

De 1,000,000 dimensiones ~ 408.25

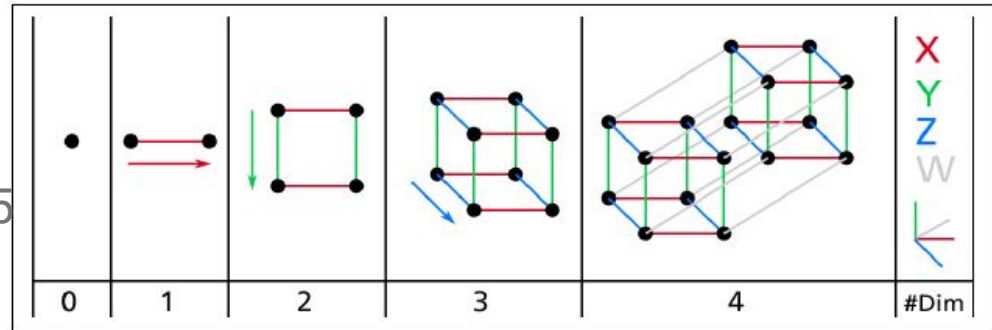
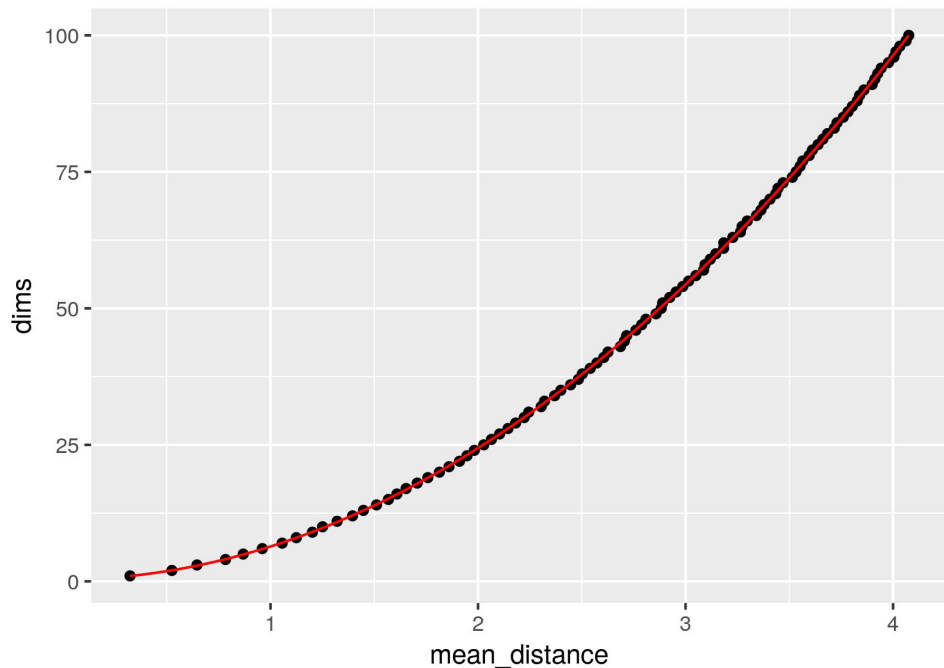


Figure 8-1. Point, segment, square, cube, and tesseract (0D to 4D hypercubes)²

La maldición de las dimensiones



$$\text{Dim} = 6x^2$$

La maldición de las dimensiones

Solo con 100 características necesitaríamos más datos que átomos en el universo para que los puntos estén concentrados a distancia 0.1

- Dispersión (díficil agrupamiento)
- Outliers

Reducción Dimensional

- Proyección
- *Manifold* (Variedad)

Proyección

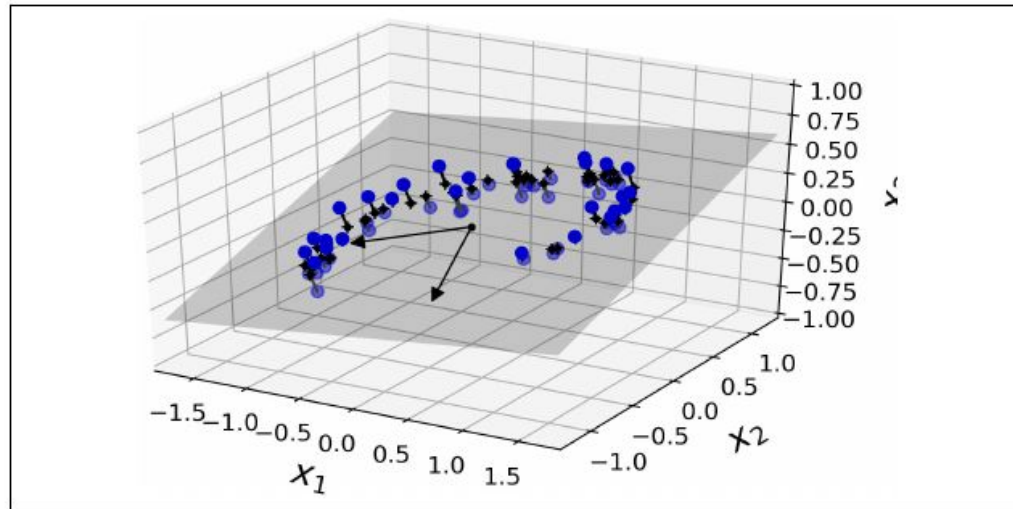


Figure 8-2. A 3D dataset lying close to a 2D subspace

Proyección

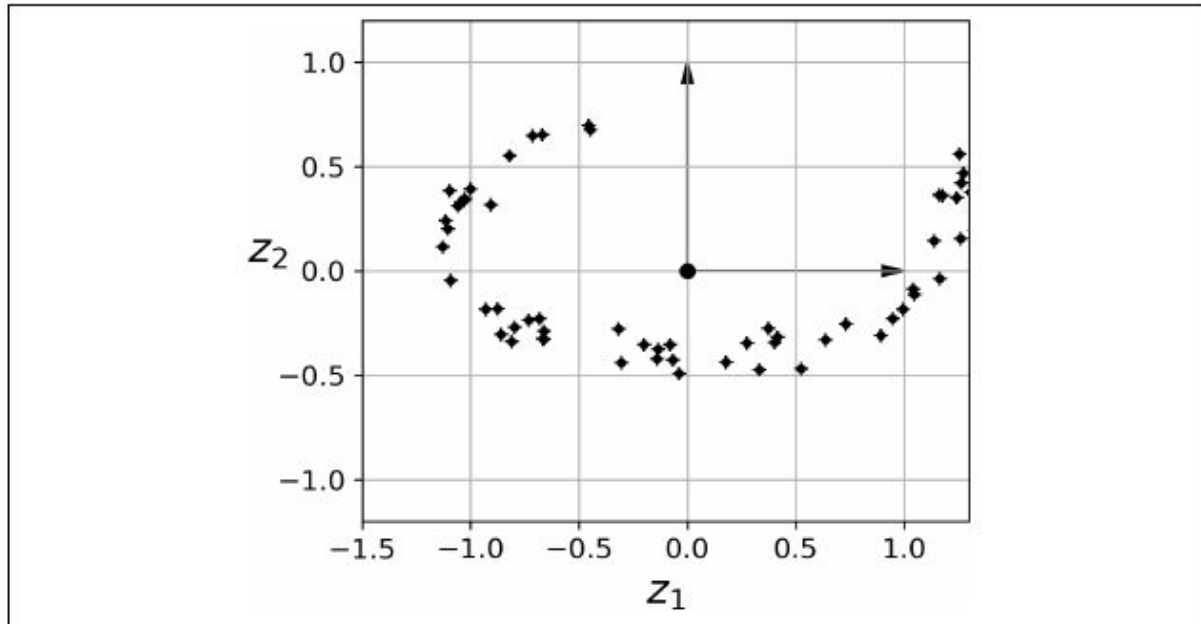


Figure 8-3. The new 2D dataset after projection

Manifold

Manifold: A grandes rasgos, una curva que puede ser doblada en dimensiones superiores. Ej. Una curva 2-D en un espacio 3-D

Buscamos un manifold 2-D que se parezca lo más posible a la forma 3-D que estamos aproximando.

La hipótesis manifold

“La mayoría de *datasets* del mundo real con muchas dimensiones se encuentran cerca de o pueden ser aproximados relativamente bien por un espacio de menor dimensión”

Manifold

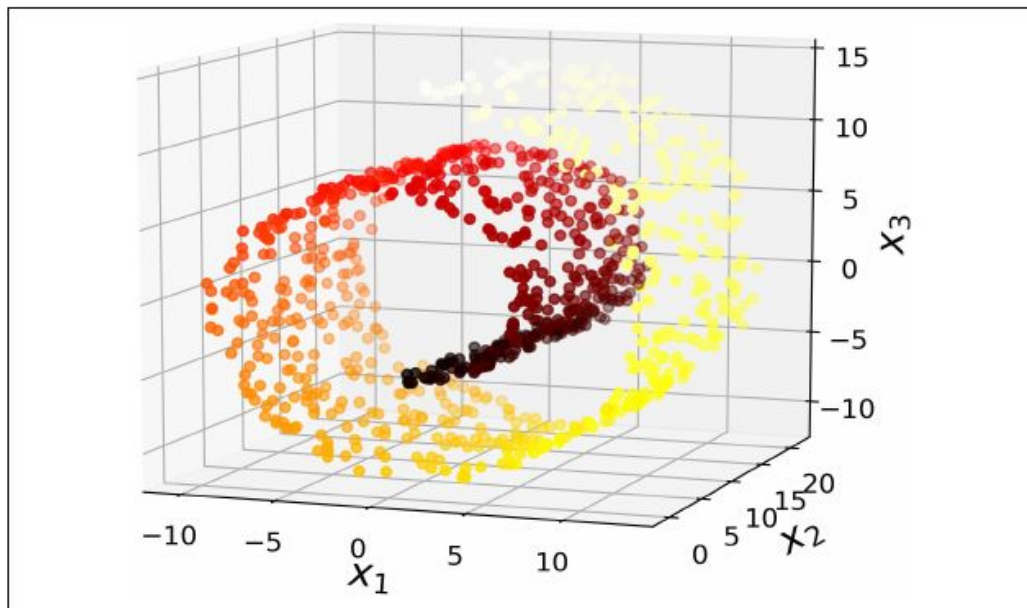


Figure 8-4. Swiss roll dataset

Manifold

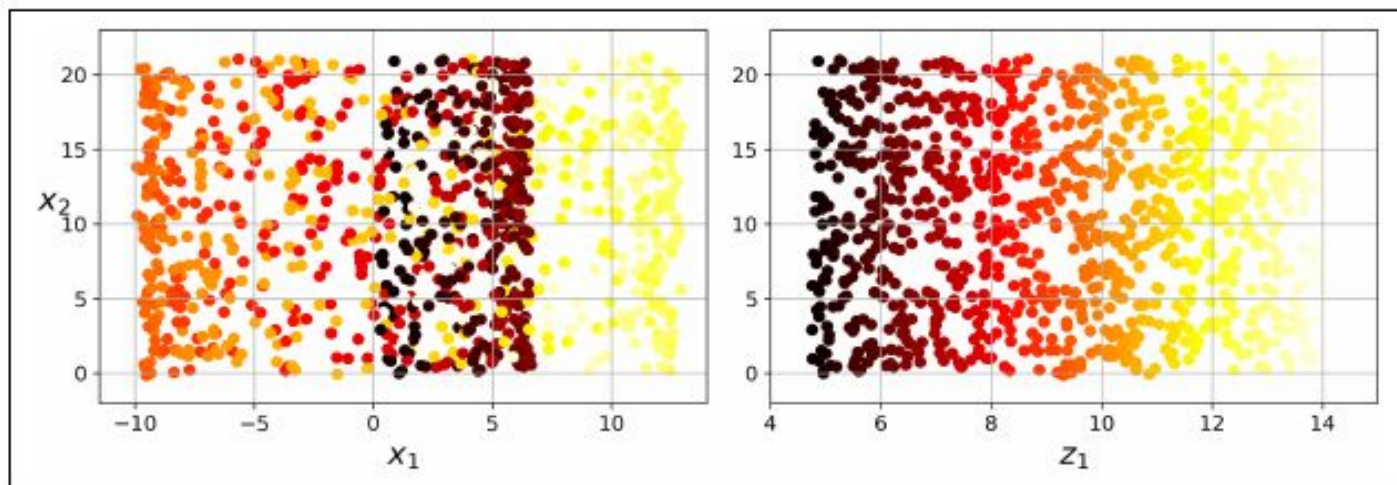


Figure 8-5. Squashing by projecting onto a plane (left) versus unrolling the Swiss roll (right)

PCA

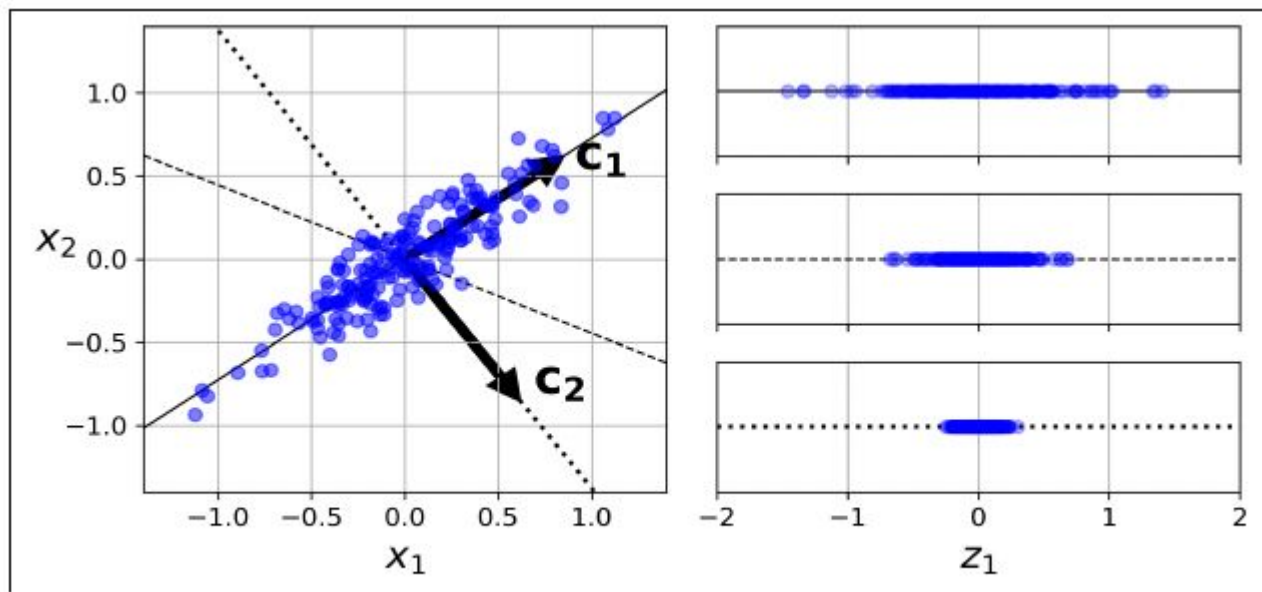


Figure 8-7. Selecting the subspace onto which to project

PCA

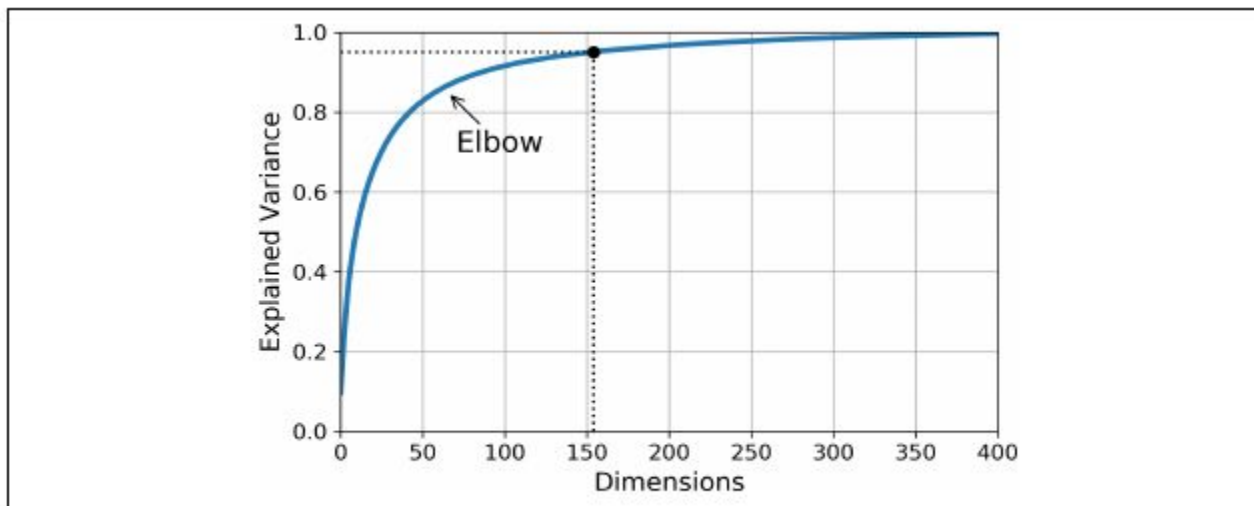


Figure 8-8. Explained variance as a function of the number of dimensions

PCA

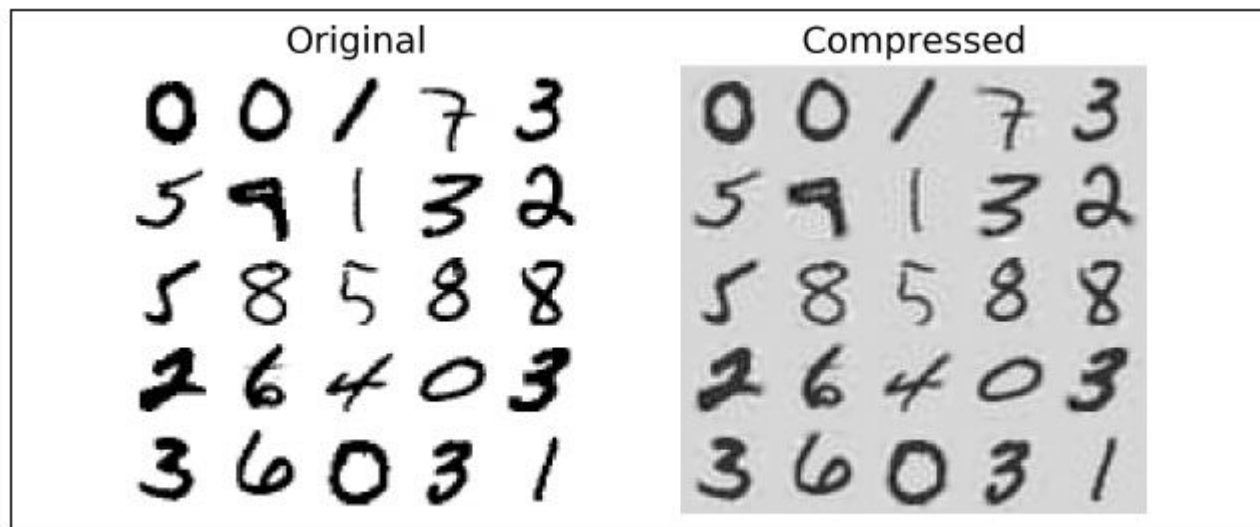
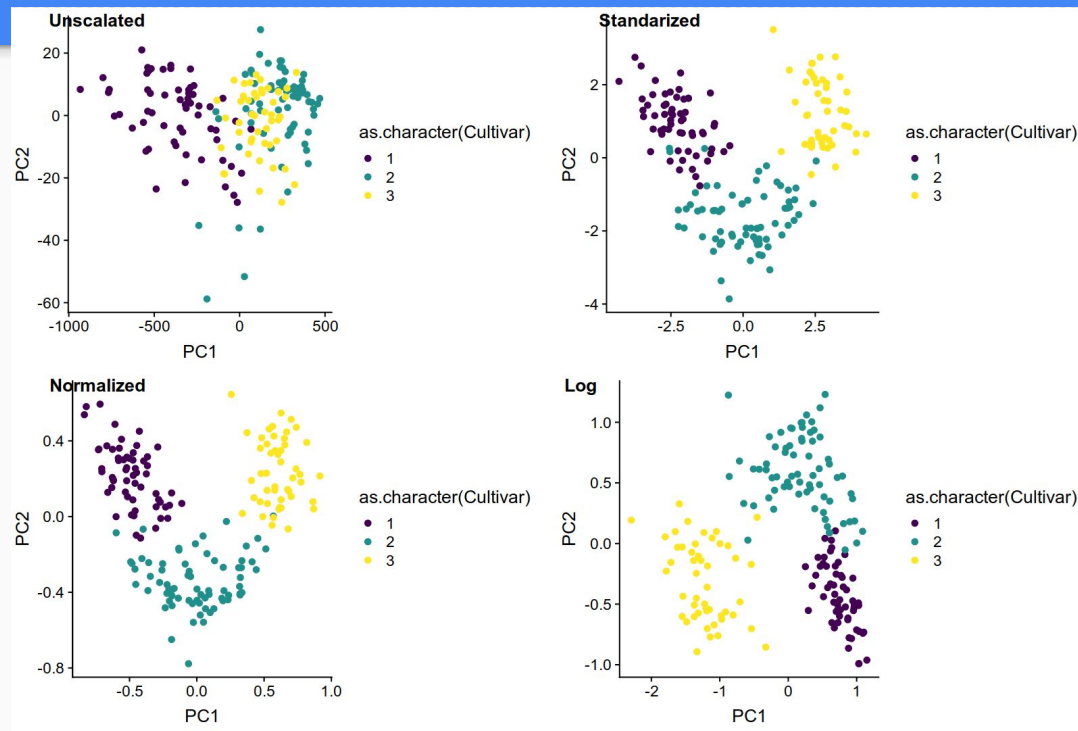


Figure 8-9. MNIST compression preserving 95% of the variance

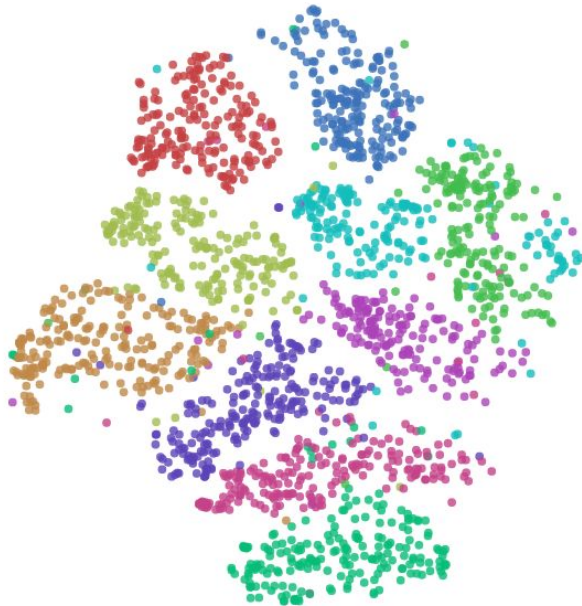
Visualización

- “Vivimos en 3-D”
- “... nos comunicamos en 2-D”
- “Y pensamos con atajos”

Ejemplo: Visualización Vinos



Ejemplo: Visualización web



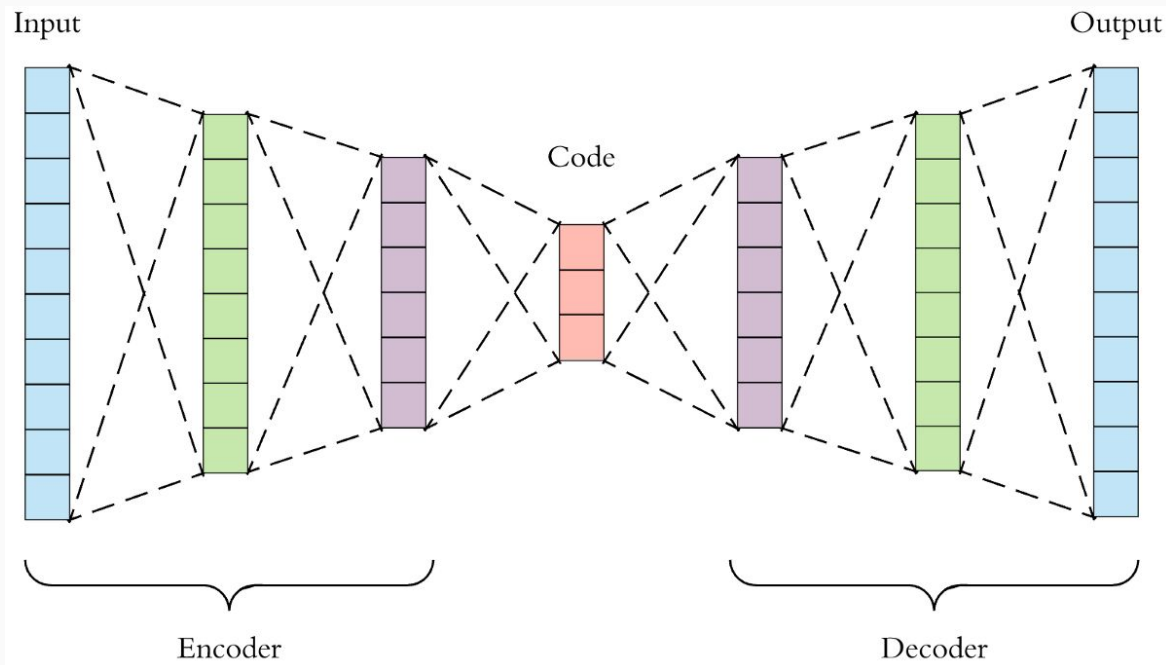
A t-SNE plot of MNIST

<http://colah.github.io/posts/2014-10-Visualizing-MNIST/>

Otras técnicas

- *Multidimensional Scaling* (MDS) reduces dimensionality while trying to preserve the distances between the instances (see Figure 8-13).
- *Isomap* creates a graph by connecting each instance to its nearest neighbors, then reduces dimensionality while trying to preserve the geodesic distances⁹ between the instances.
- *t-Distributed Stochastic Neighbor Embedding* (t-SNE) reduces dimensionality while trying to keep similar instances close and dissimilar instances apart. It is mostly used for visualization, in particular to visualize clusters of instances in high-dimensional space (e.g., to visualize the MNIST images in 2D).
- *Linear Discriminant Analysis* (LDA) is actually a classification algorithm, but during training it learns the most discriminative axes between the classes, and these axes can then be used to define a hyperplane onto which to project the data. The benefit is that the projection will keep classes as far apart as possible, so LDA is a good technique to reduce dimensionality before running another classification algorithm such as an SVM classifier.

Autoencoder



Prácticas

PCA

<https://drive.google.com/open?id=1kSa-YsE2ISA5FnZzGGxlbJlWBF1yLV3X>

Patrones frecuentes

https://drive.google.com/open?id=1Am3XehBx_1ojJBMxvxkbagRtduB8ZcuM

Patrones frecuentes

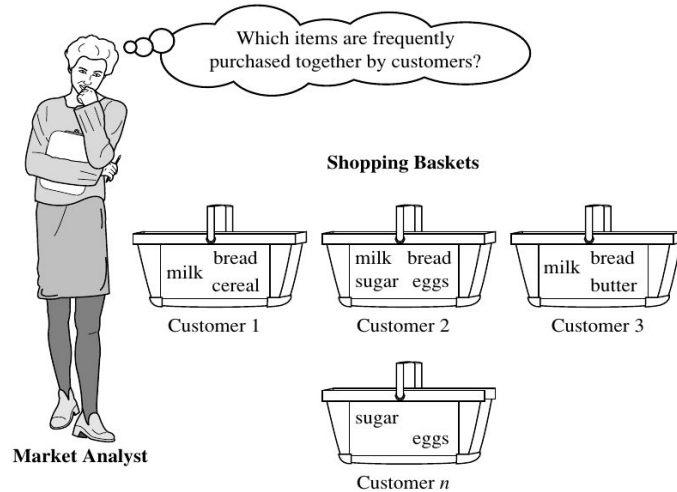


Figure 6.1 Market basket analysis.

Patrones frecuentes

Encontrar:

1. Canastas frecuentes
2. Reglas sólidas de canastas frecuentes



Patrones frecuentes

computer \Rightarrow antivirus_software [support = 2%, confidence = 60%].

Reglas sólidas

cumplen:

- Con soporte mínimo
- Confianza mínima

$$\text{support}(A \Rightarrow B) = P(A \cup B)$$

$$\text{confidence}(A \Rightarrow B) = P(B|A).$$

$$\text{confidence}(A \Rightarrow B) = P(B|A) = \frac{\text{support}(A \cup B)}{\text{support}(A)} = \frac{\text{support_count}(A \cup B)}{\text{support_count}(A)}.$$

Algoritmo Apriori

1. Generación de Candidatos (*join*) $L_{k-1} \rightarrow c_k$
2. Podado usando la propiedad Apriori C_k
3. Pasada por la base de datos para generar L_k

Propiedad Apriori: Todos los subconjuntos de compras de un conjunto frecuente, también son frecuentes.

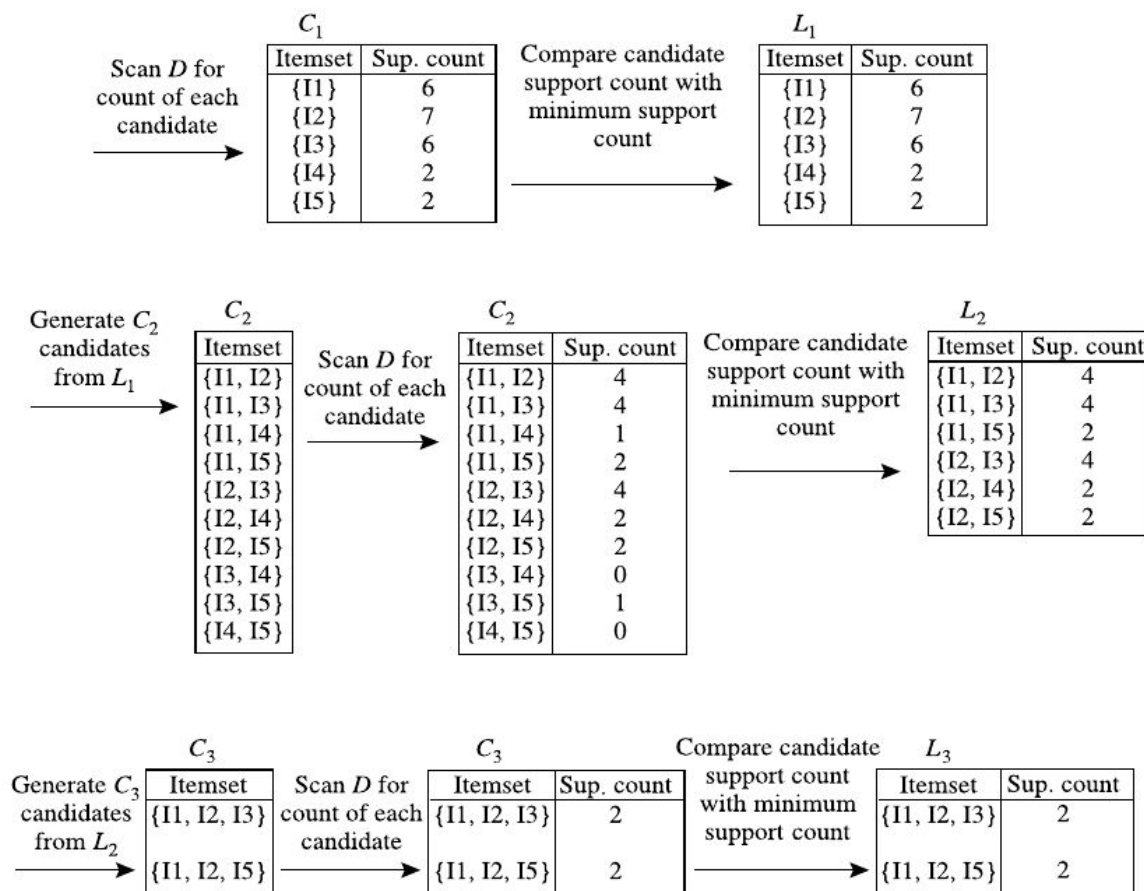


Figure 6.2 Generation of the candidate itemsets and frequent itemsets, where the minimum support count is 2.