



Clúster

APRENDIZAJE NO SUPERVISADO

Tipos de ML



Supervisado

Predicción y
etiquetado



No Supervisado

Identifica
clústeres

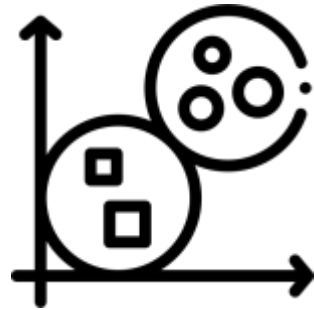


Reforzado

Aprende de
errores

Aprendizaje No Supervisado

- No existe etiqueta de salida
- El objetivo es descubrir relaciones entre la data existente

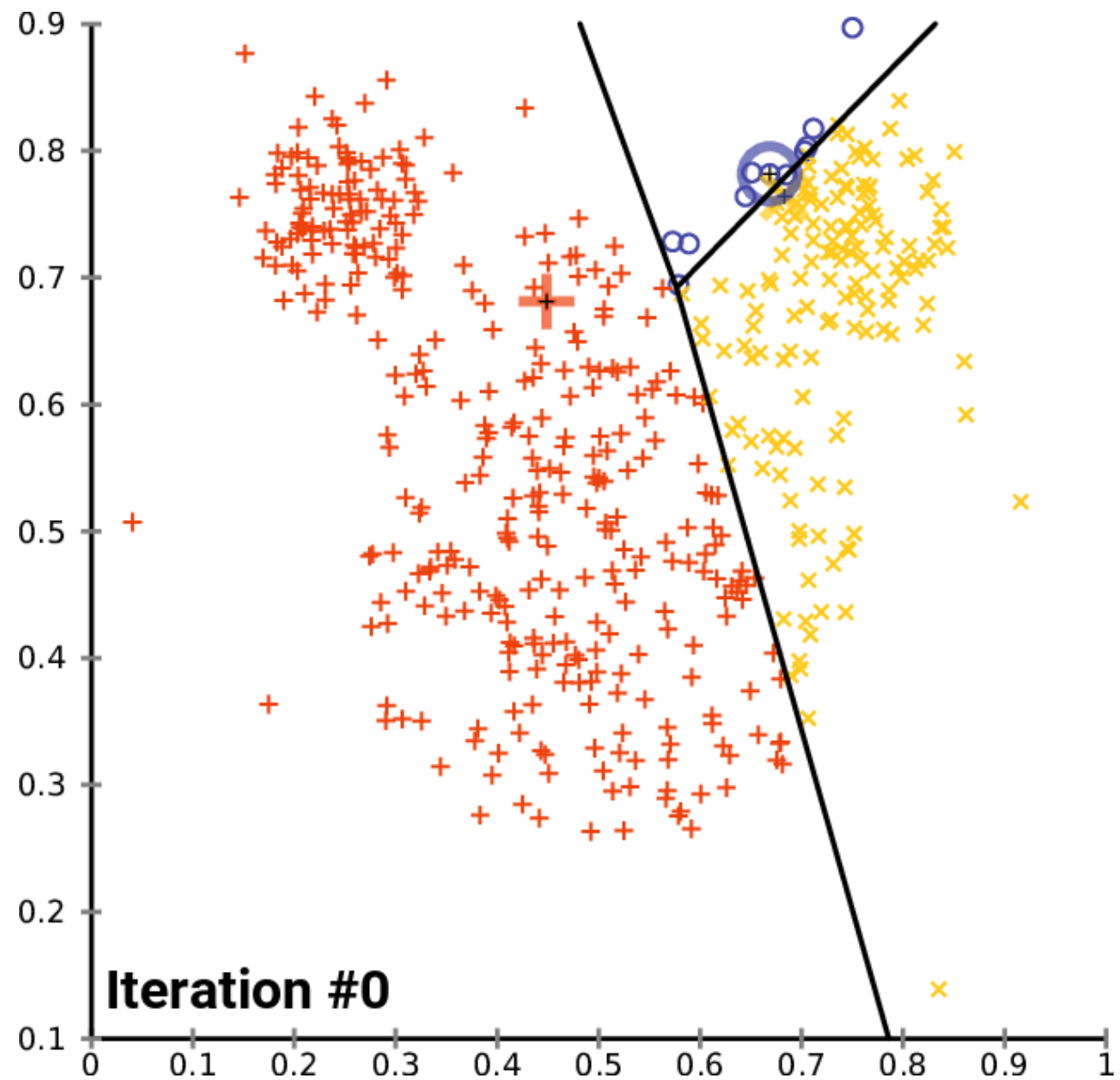


Clasificación

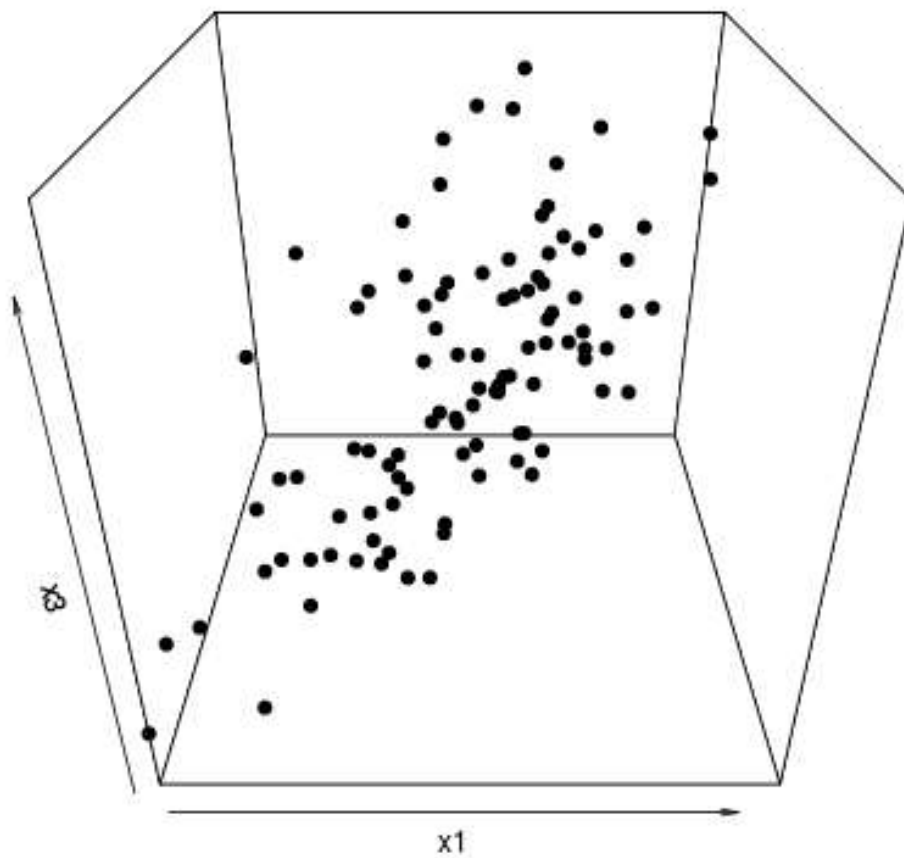


Reducción de
dimensiones

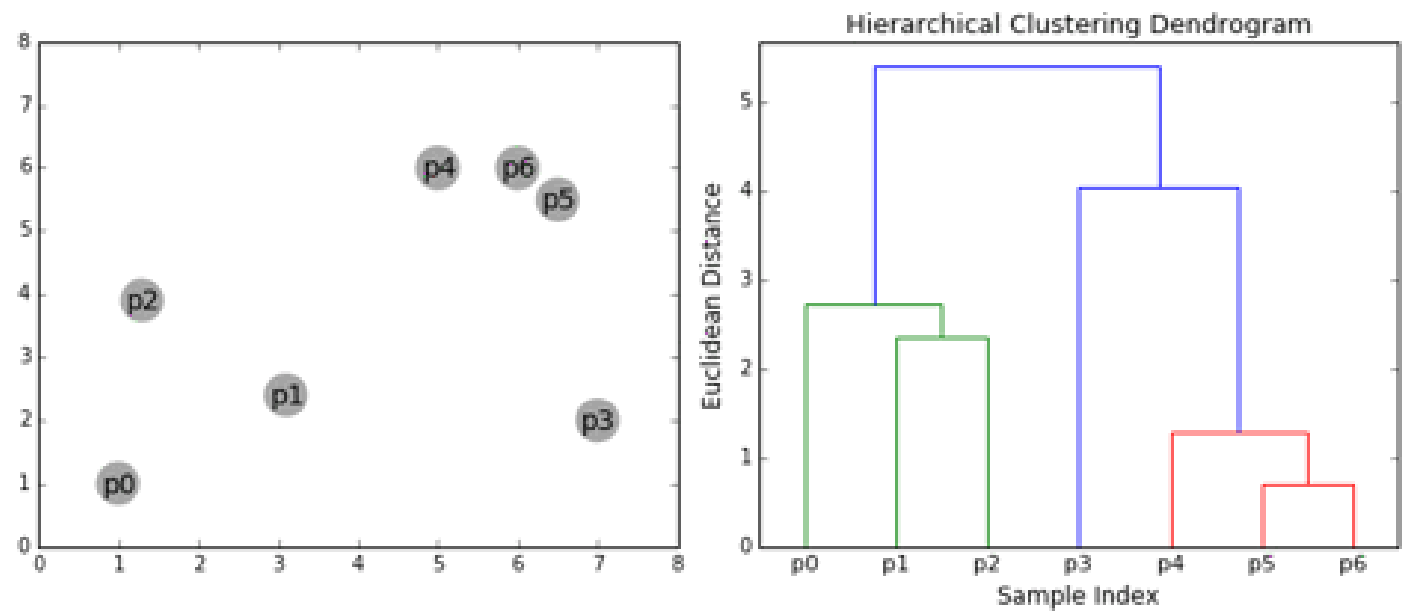
K-Means



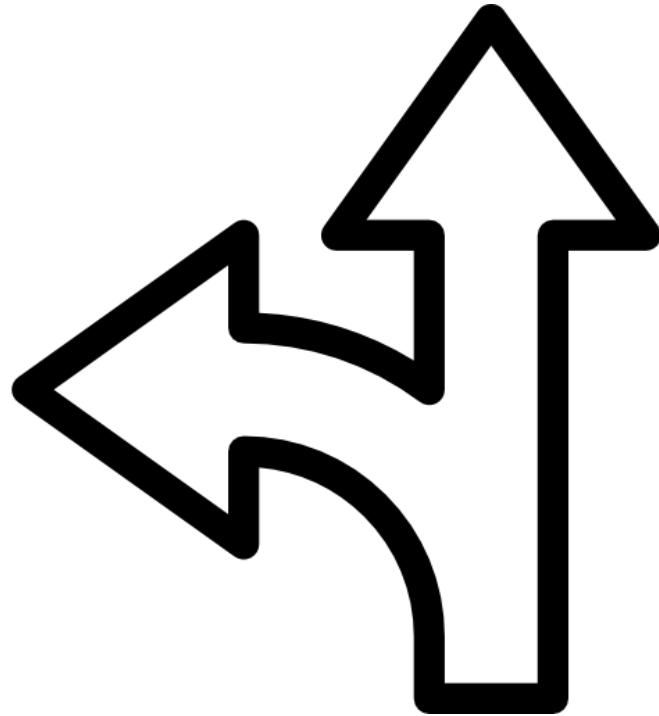
PCA



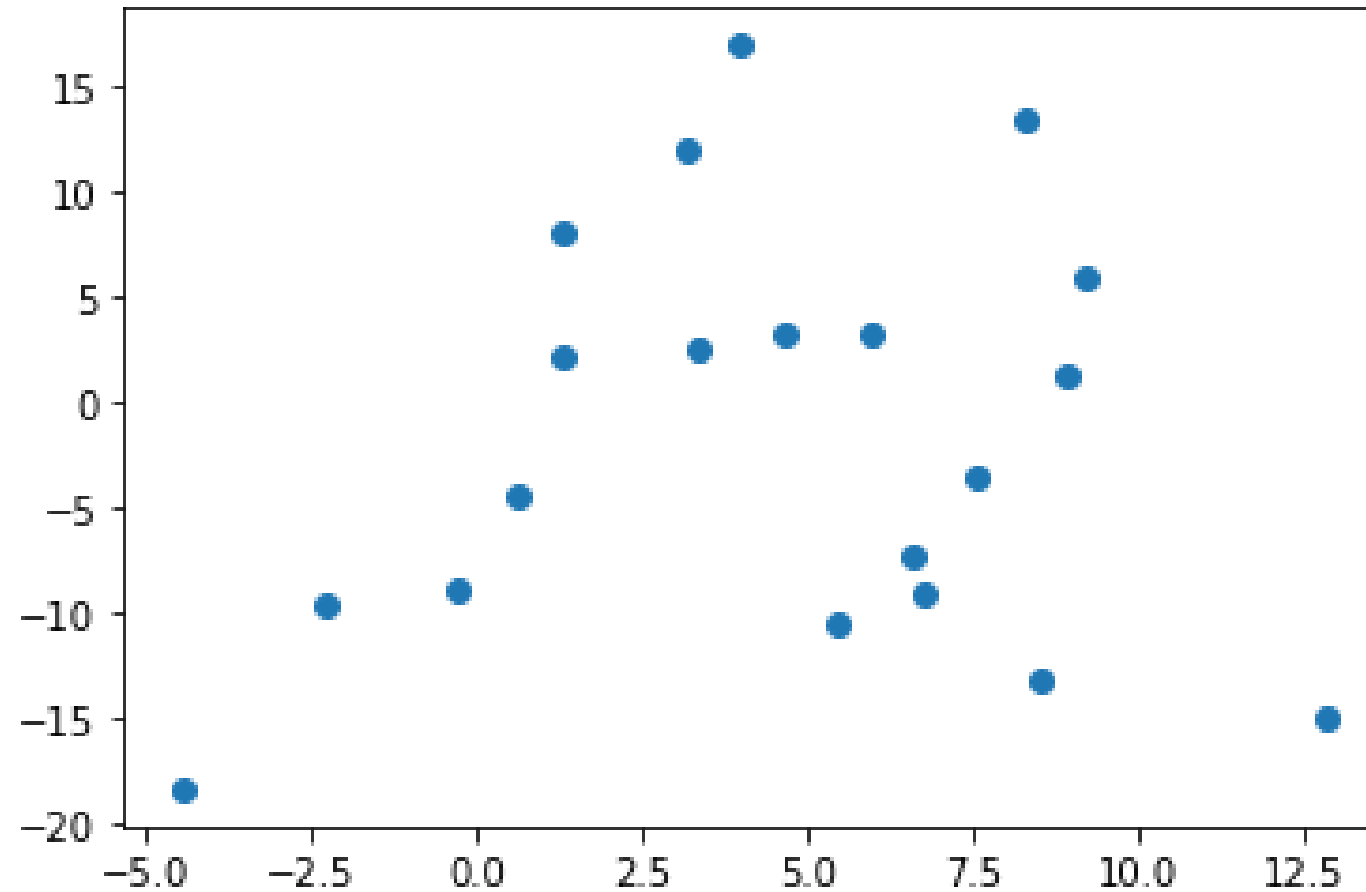
Agrupación jerárquica



Pequeño desvío de regreso a Supervisados



K-Nearest Neighbors

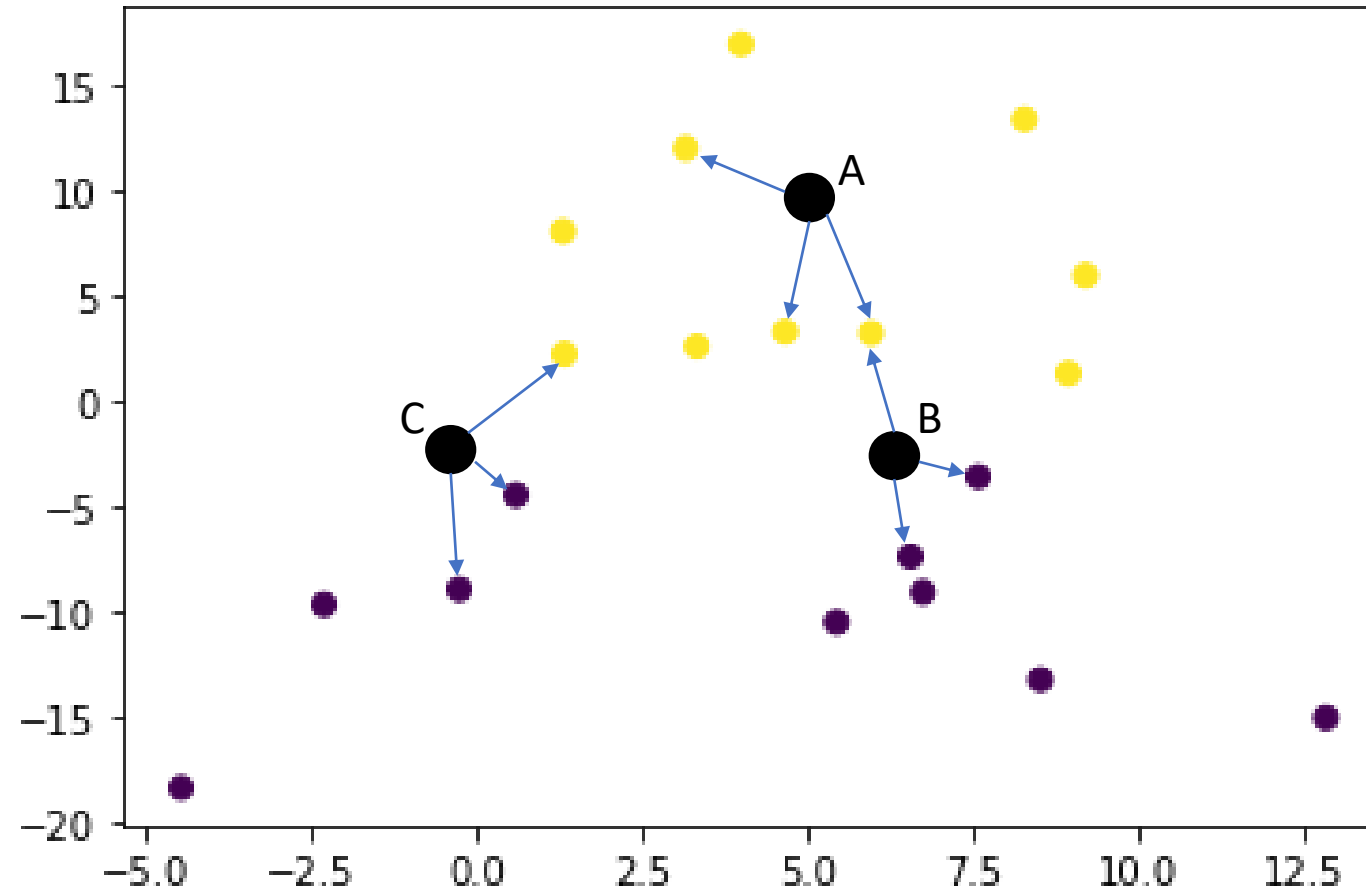




K-Nearest Neighbors

- Clasificamos por similitud
- Vecinos más cercanos
- Simple
- Dado una data desconocida, a qué clase pertenece?

K-Nearest Neighbors



$k = 3$

K-Nearest

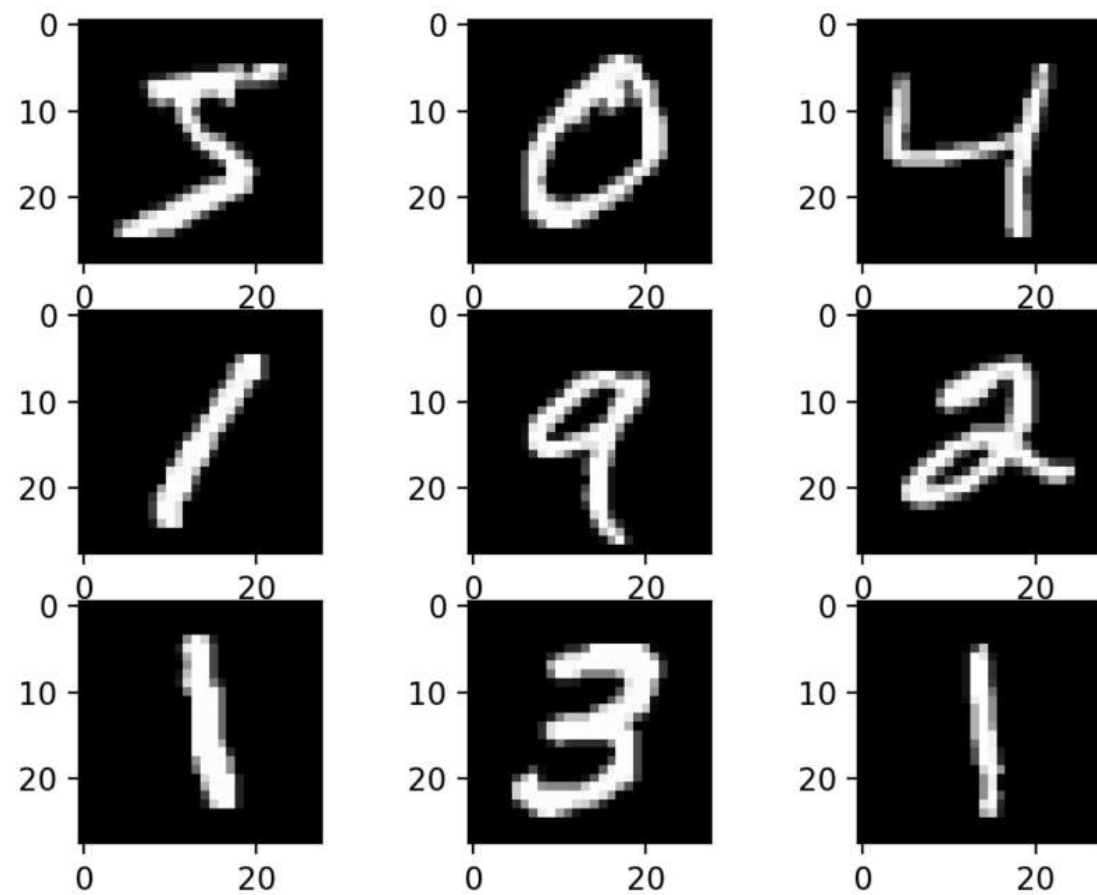
El aprendizaje está solo en la data
Predecir es costoso

- Data: Un grupo de **entrenamiento** de data etiquetada
- Data: Un grupo de **prueba** de data no etiquetada
- Data: Un entero **K**

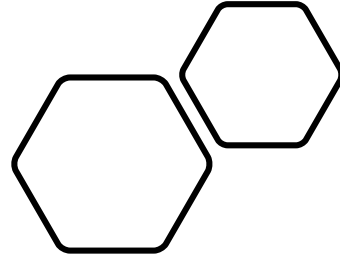
Algoritmo: Para cada ítem en prueba, obtener la etiqueta más común de los K vecinos más cercanos en la data de entrenamiento

- Por cada ítem x en prueba
 - Por cada ítem y en entrenamiento
 - Calcular similitud (x,y)
 - Encontrar los k ítems más similares a x
 - Calcular la etiqueta más común

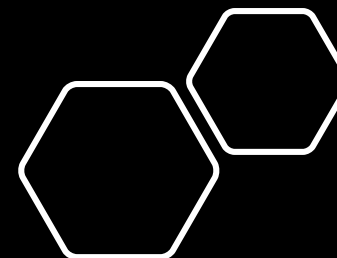
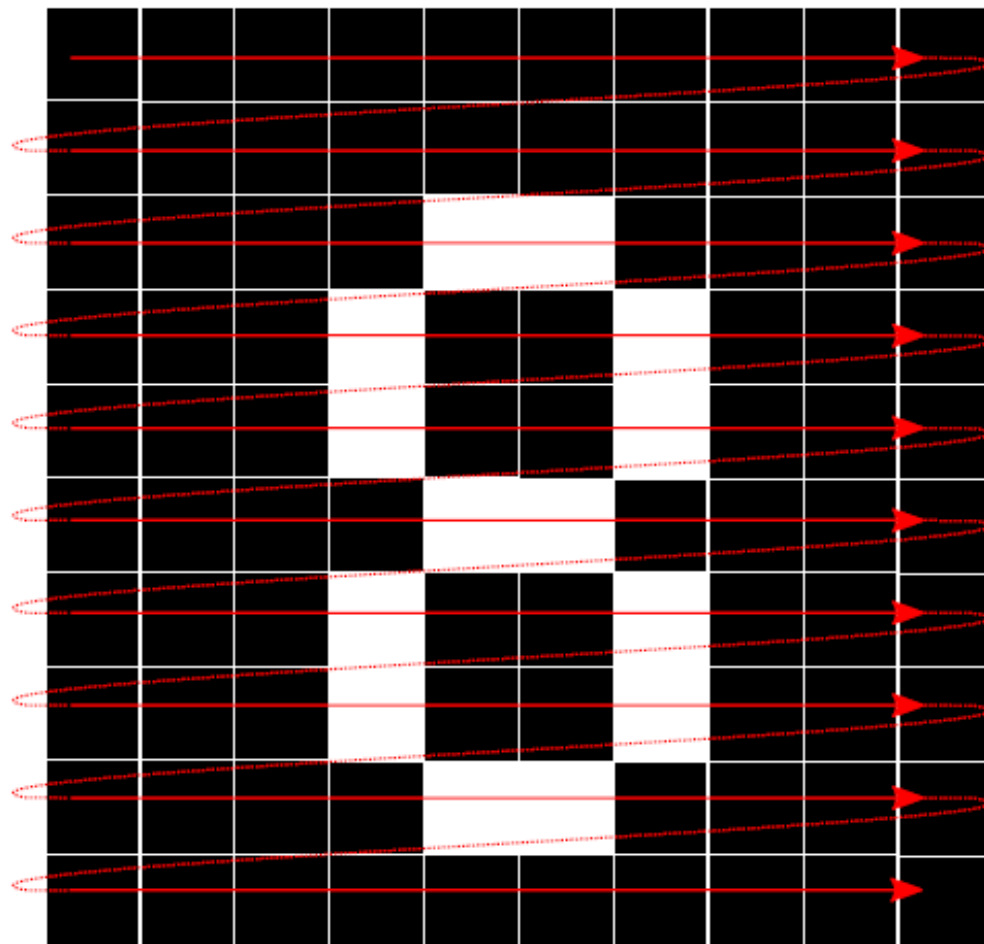
Probando
con MNIST

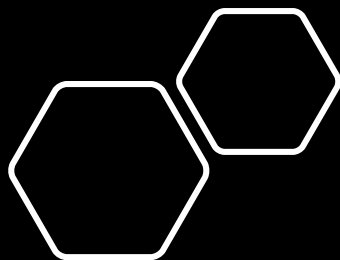
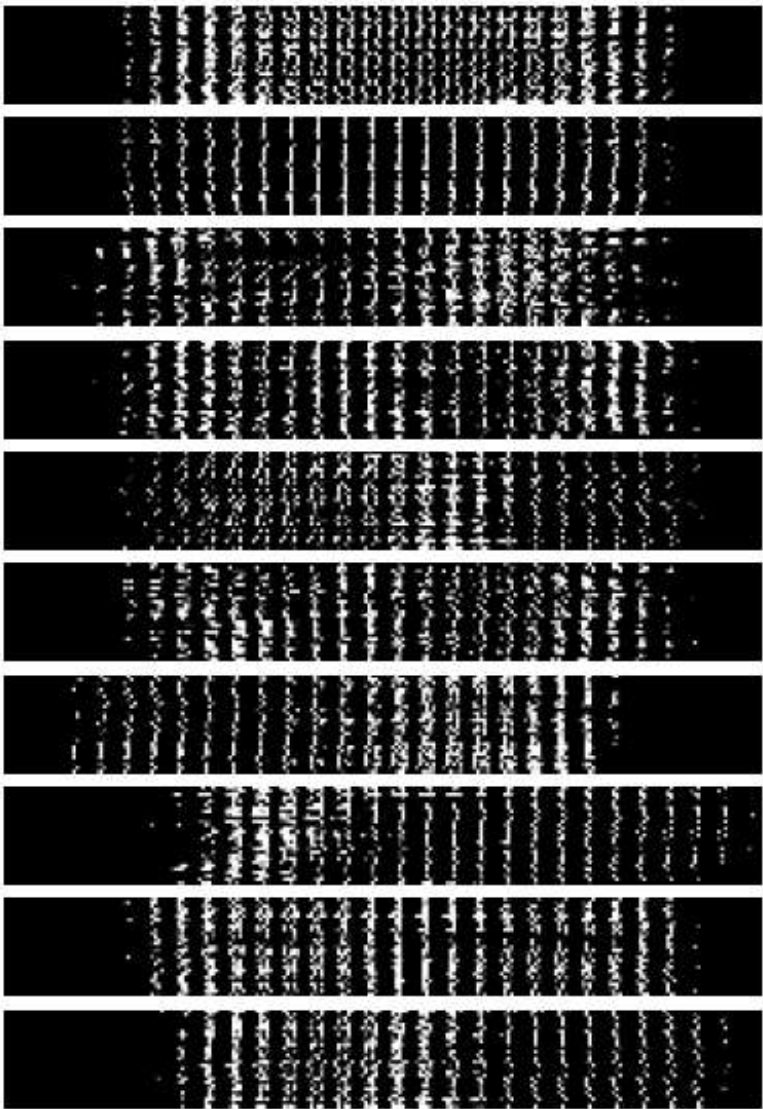


KNN y MNIST



- ¿Será bueno?
- Hay bastante variabilidad entre números
- ¿Similitud?
 - Euclidiana
- Más pequeño, más parecido



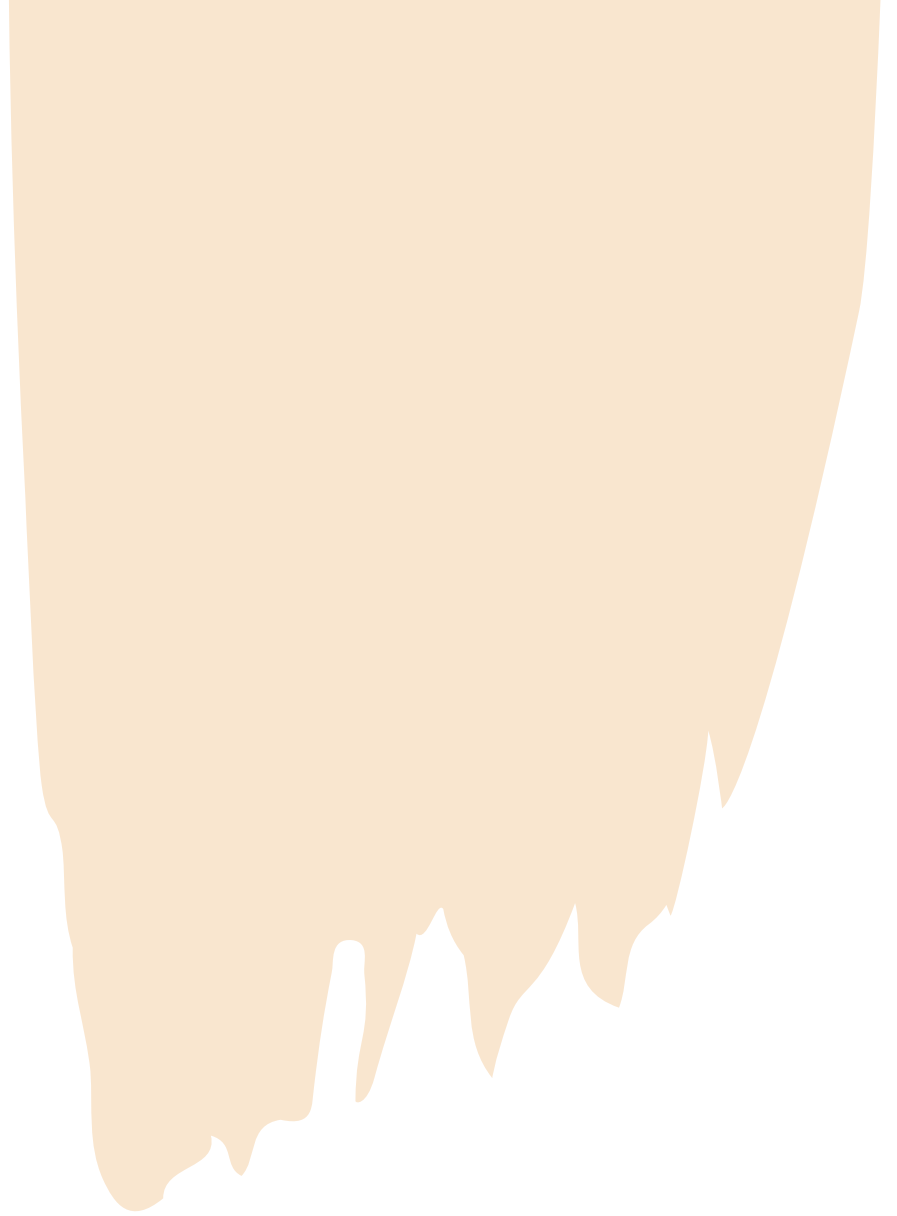


DEMO



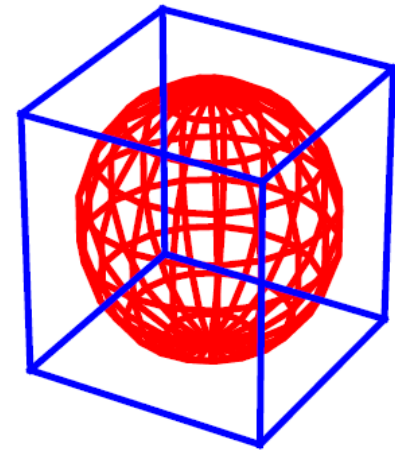
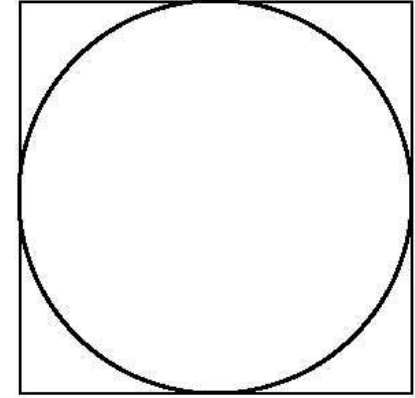
MNIST_KNN_DEMO.ipynb

Maldición de la dimensión



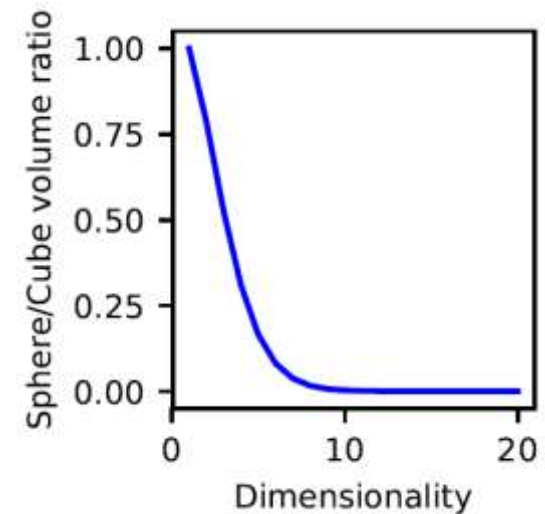
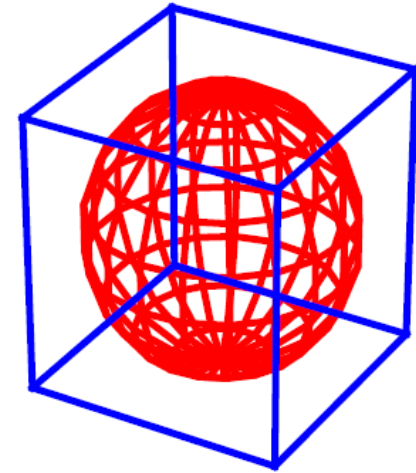
Espacios en altas dimensiones


- Espacios en altas dimensiones tienen propiedades extrañas
- Hipercubo de n dimensiones con lado $2r$
- Hiperesfera de n dimensiones dentro del hipercubo de tal forma que el la superficie intercepta las “caras” del hipercubo



Relación del volumen del cubo vs la esfera

- En 2D, la esquina de un cuadrado es $\sqrt{2}r \approx 1.41r$ del centro
- En 3D, la esquina de un cubo es $\sqrt{3}r \approx 1.73r$ del centro.
- En 4D, $2r$; y en 5D, $2.23r$
- En 1000D, las esquinas del hipercubo se encuentran 30 veces más lejos de la hiperesfera que encierra.

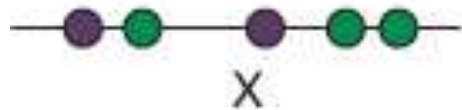




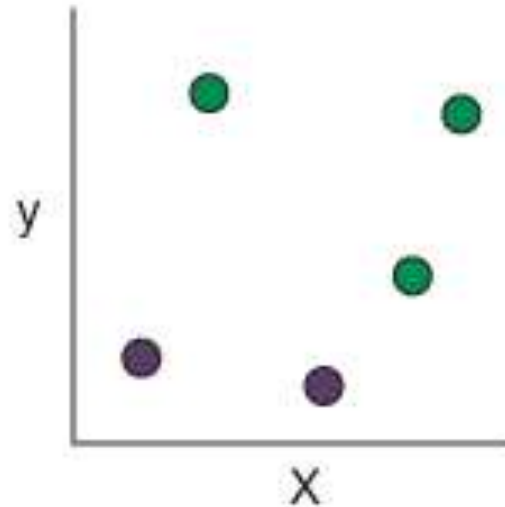
*¿Qué tiene
que ver con
la data?*

En altas dimensiones

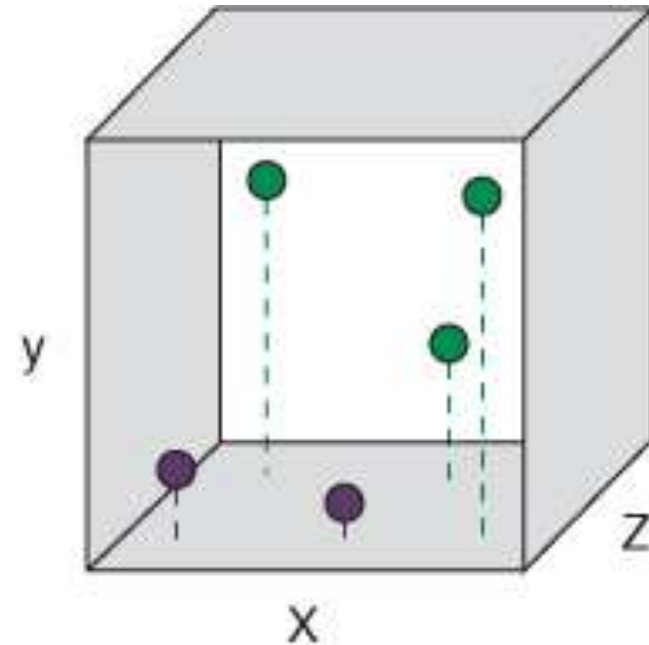
- Mientras más dimensiones, la data estará más espaciada
- Más ejemplos de nuestra data vamos a requerir



One dimension



Two dimensions



Three dimensions

Ejemplo Binario

Tener al menos 10 ejemplos de cada “clase”

1 Feature Binario (1D)

- $[0,1] \rightarrow 0 \ 1$
- 20 ejemplos en total

Ejemplo Binario

Tener al menos 10 ejemplos de cada “clase”

2 Features Binarios (2D)

- $[0,1] [0,1] \rightarrow 00\ 01\ 10\ 11$
- 40 ejemplos en total

Ejemplo Binario

Tener al menos 10 ejemplos de cada “clase”

3 Features Binarios (3D)

- $[0,1] [0,1] [0,1] \rightarrow 000\ 001\ 010\ 011\ 100\ 101\ 110\ 111$
- 80 ejemplos en total

Ejemplo Binario

Tener al menos 10 ejemplos de cada “clase”

k Features Binarios (kD)

- 2^k combinaciones únicas
- Se necesitan $10 * 2^k$ ejemplos
- Para 17 features, se necesitan 1M de ejemplos
- Para 20, requieres de 10M



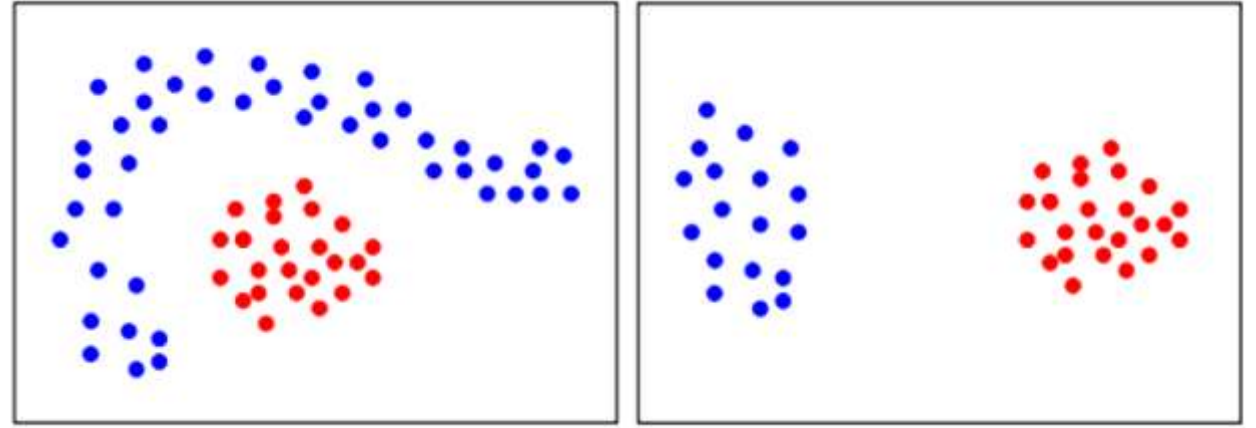
Solución

Reducción de Dimensiones

Trabajar con solo lo necesario

Ingeniería de Features

PCA



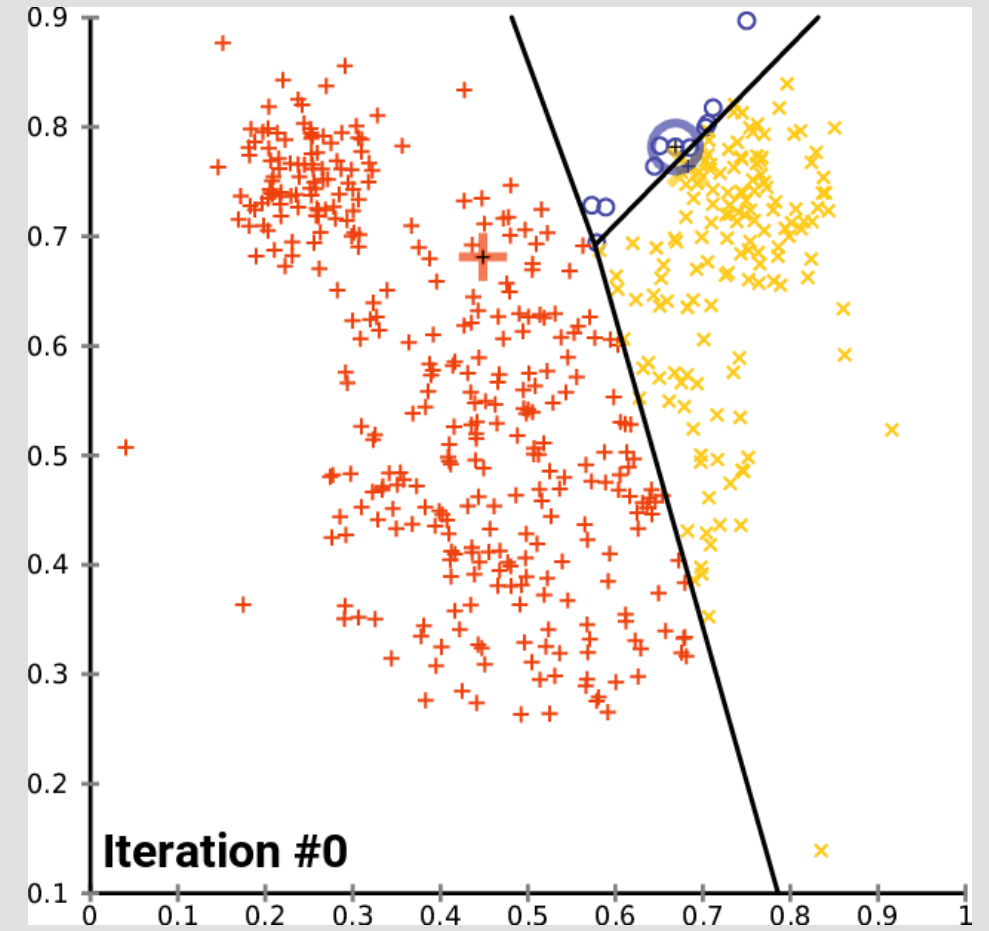
Clustering

Idea General de Clustering

- Varios algoritmos de clustering
- Muchos asumen algo sobre la data, ej.. la data en los clúster está normal
- Muchos agrupan puntos que son “similares”
- La salida es un grupo de puntos que “son del grupo”

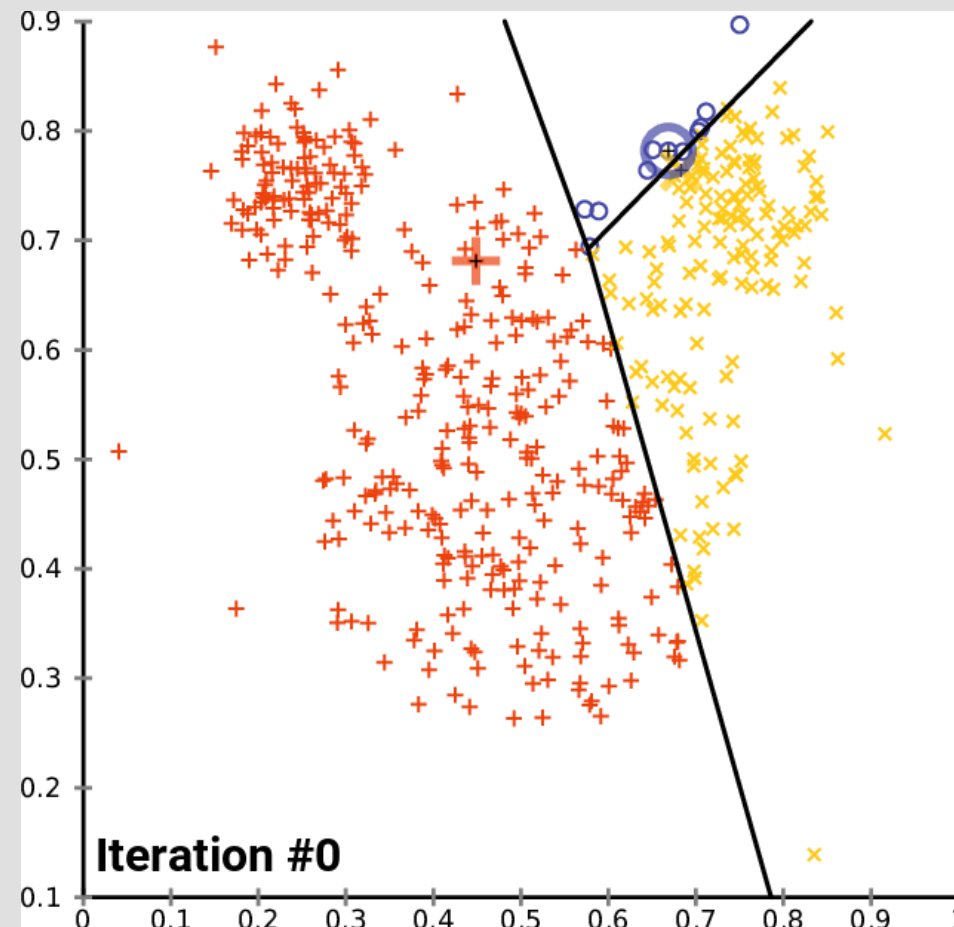
K-Means

- K-Medios
- Algoritmo no supervisado
- Iterativo
- El algoritmo más común



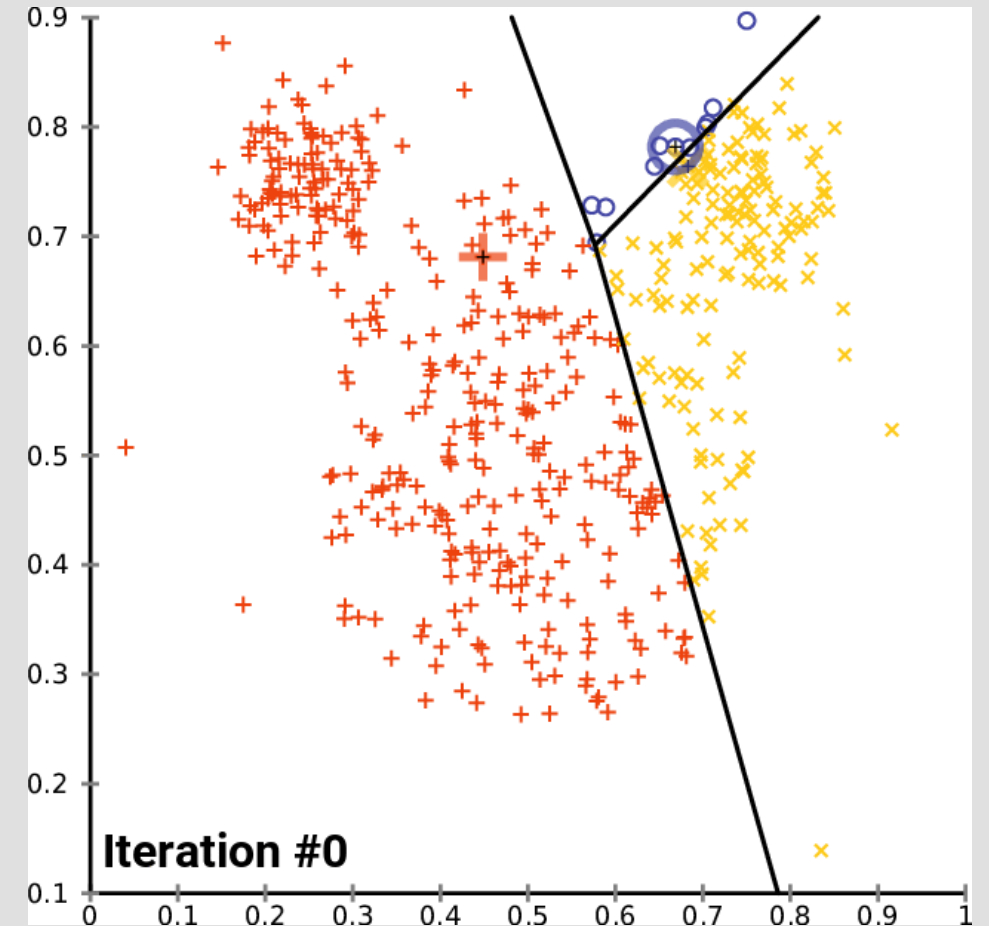
K-Means

- Aprende un Vector prototipo
- Asigna todos los puntos al vector más cercano
- El espacio es dividido en celdas, uno por prototipo
- El vector prototipo es también conocido como centroide



K-Means

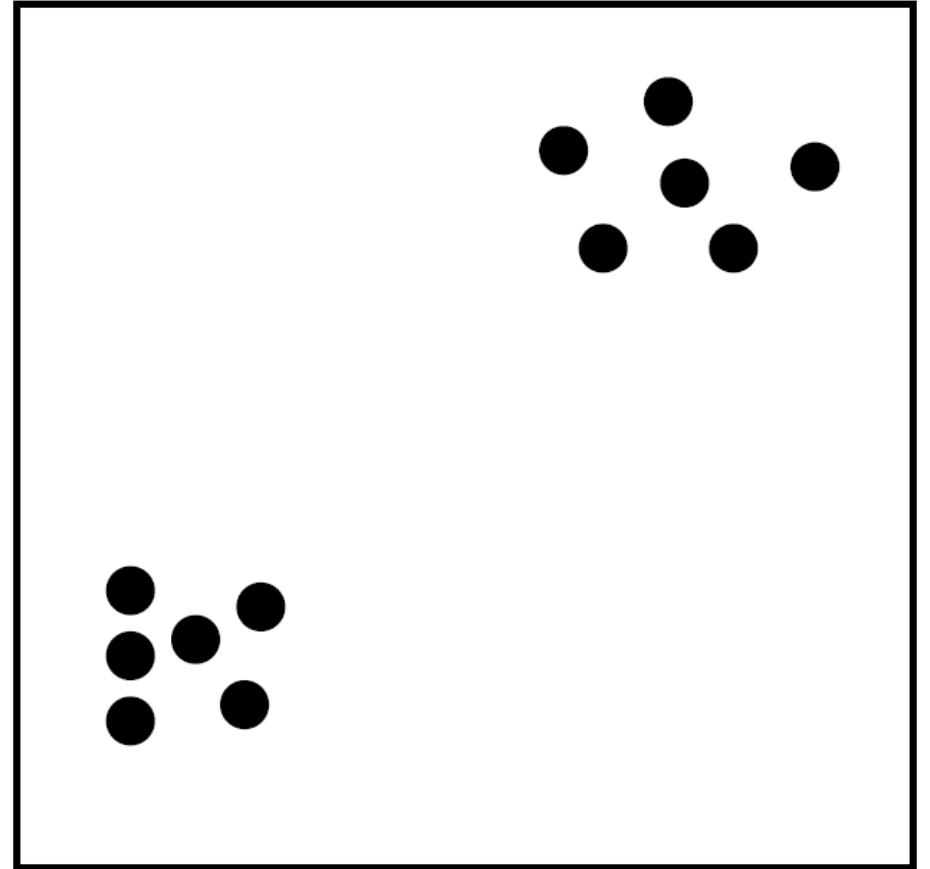
- Es un algoritmo paramétrico
- Asume varias cosas sobre la data
- El numero k de clúster debe ser especificado



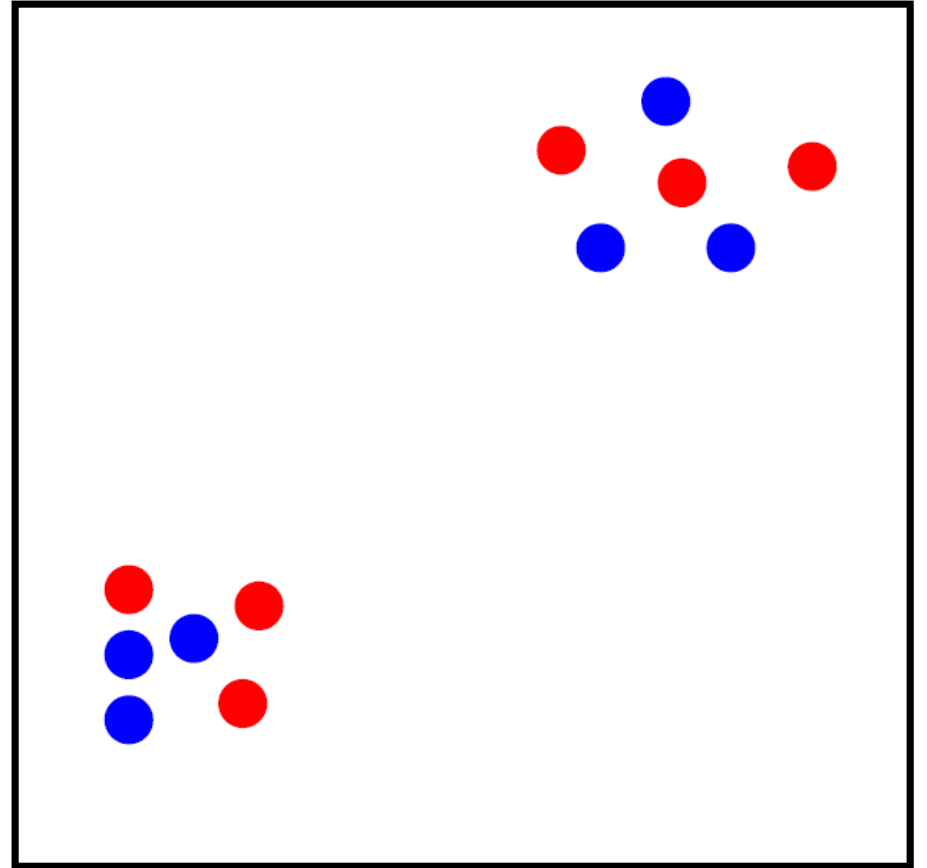
K means (algoritmo)

1. Asignar aleatoriamente cada punto a un clúster
2. Computar los centroides (posición promedio) de cada clúster
3. Por cada punto, calcular la distancia al centroide de cada clúster
4. Reasignar el punto al clúster con el centroide más cercano
5. Regresar a 2 y repetir hasta no haber cambios en los clúster

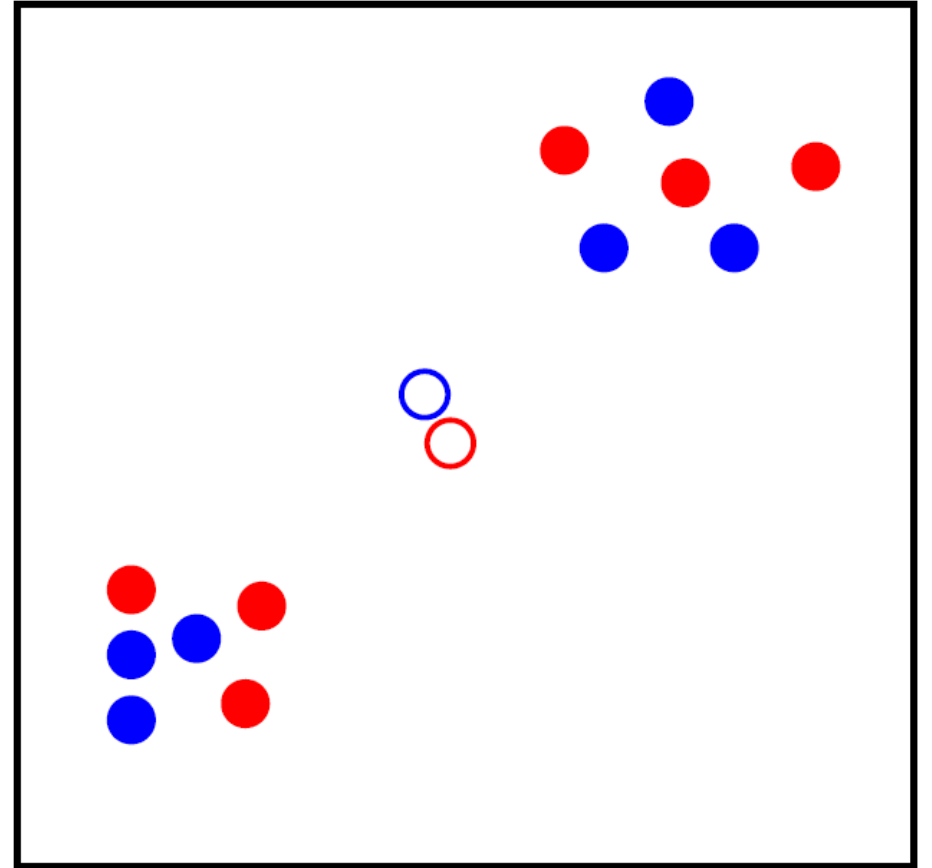
Situación Inicial



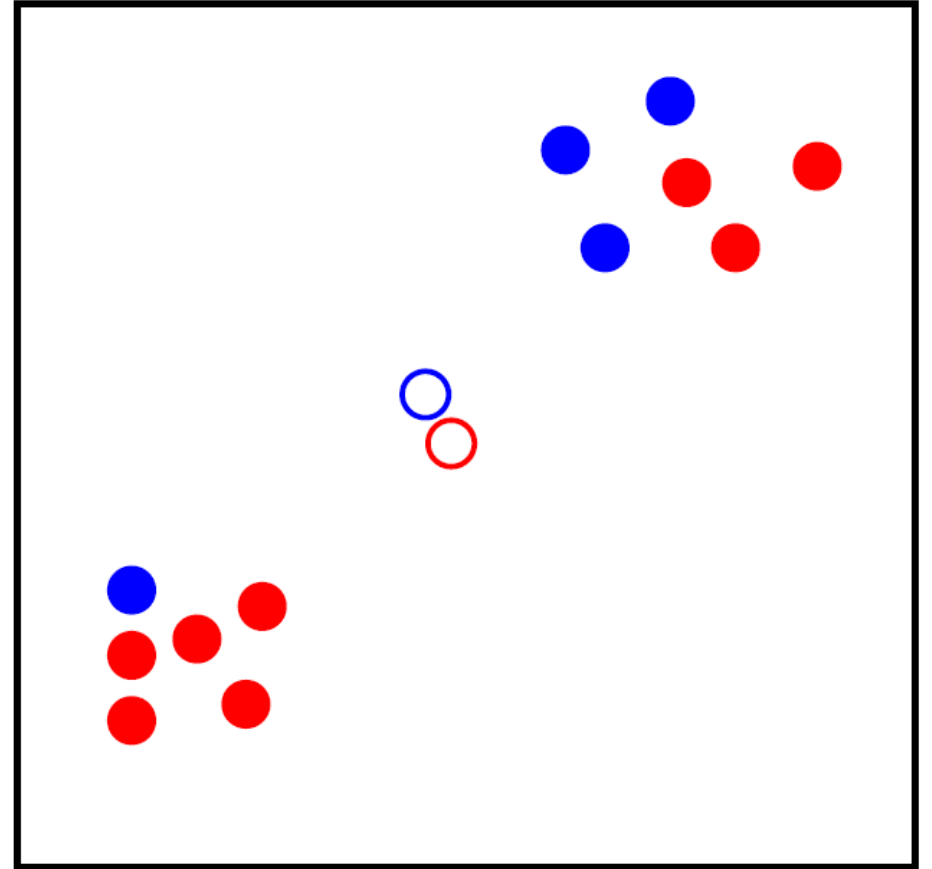
*Asignar aleatoriamente
grupos*



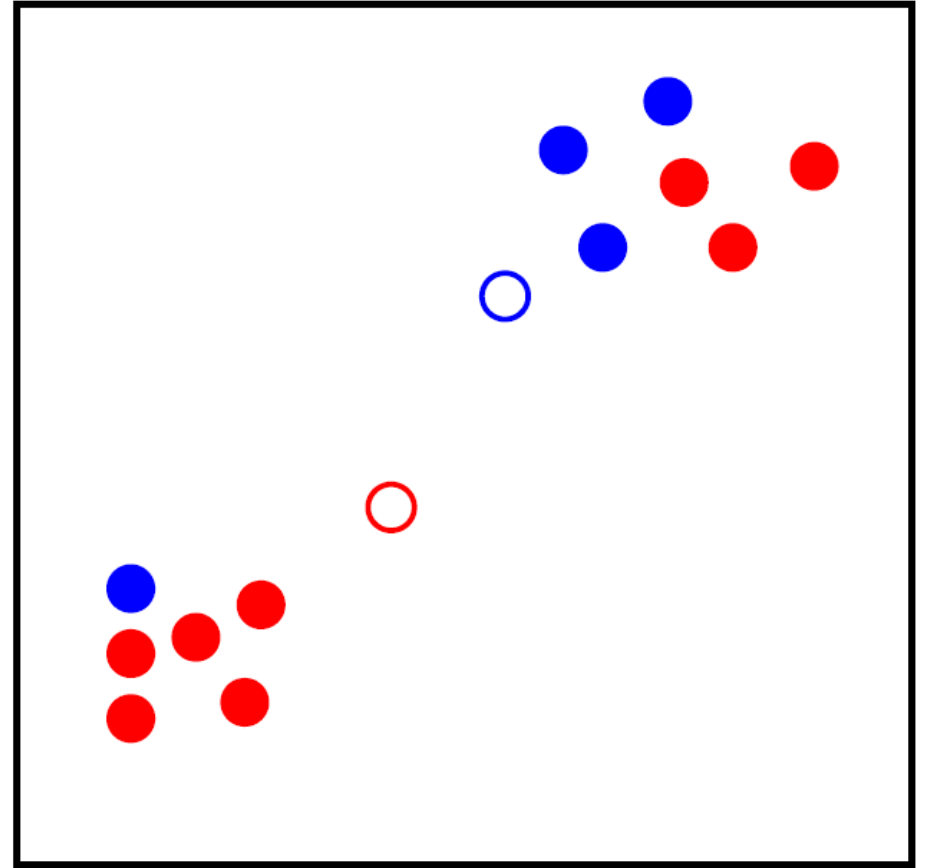
Calcular los centroides



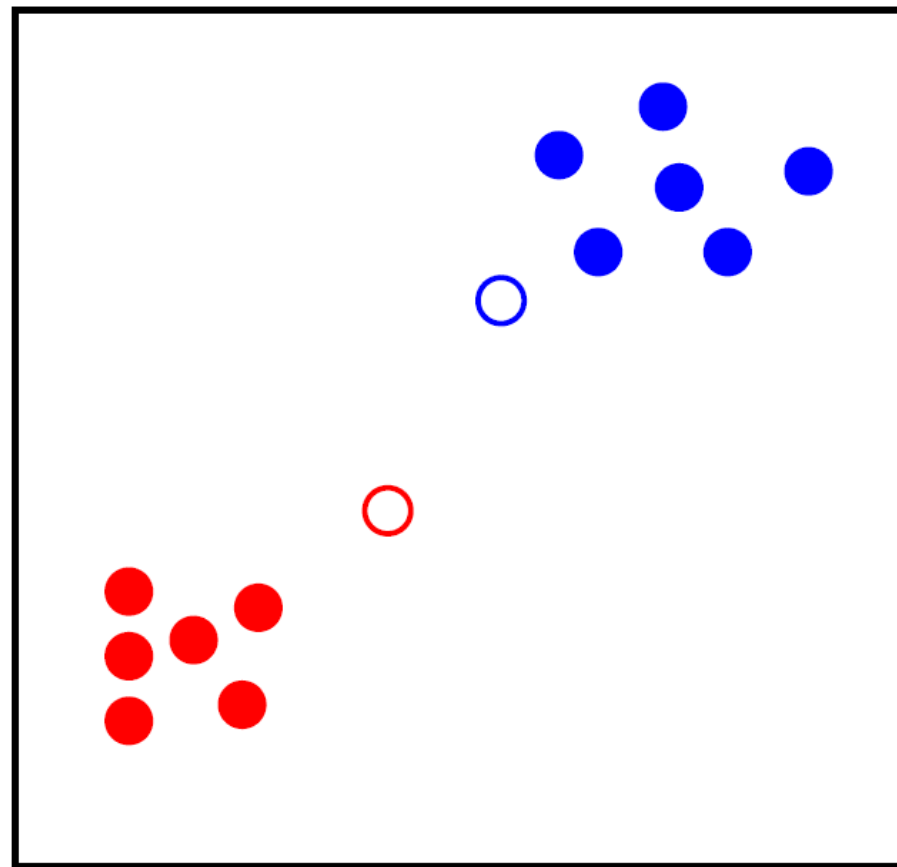
Reasignamos los grupos



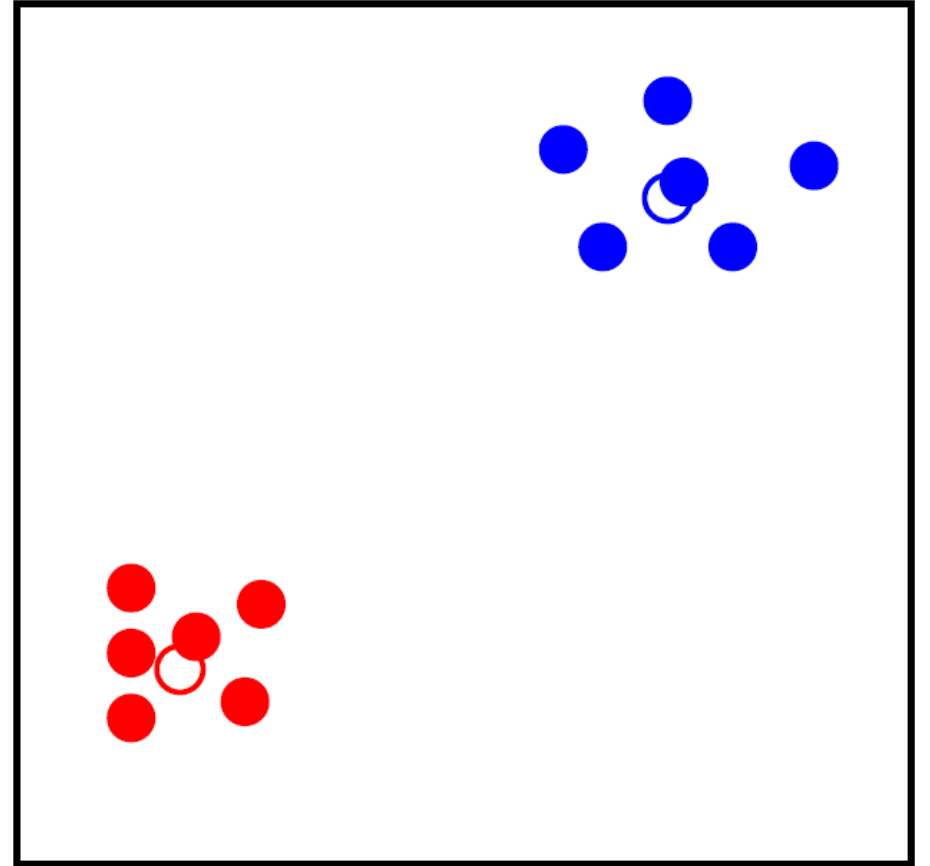
*Volvemos a calcular los
centroides*



*Volvemos a reasignar
los grupos*



Volvemos a calcular el promedio

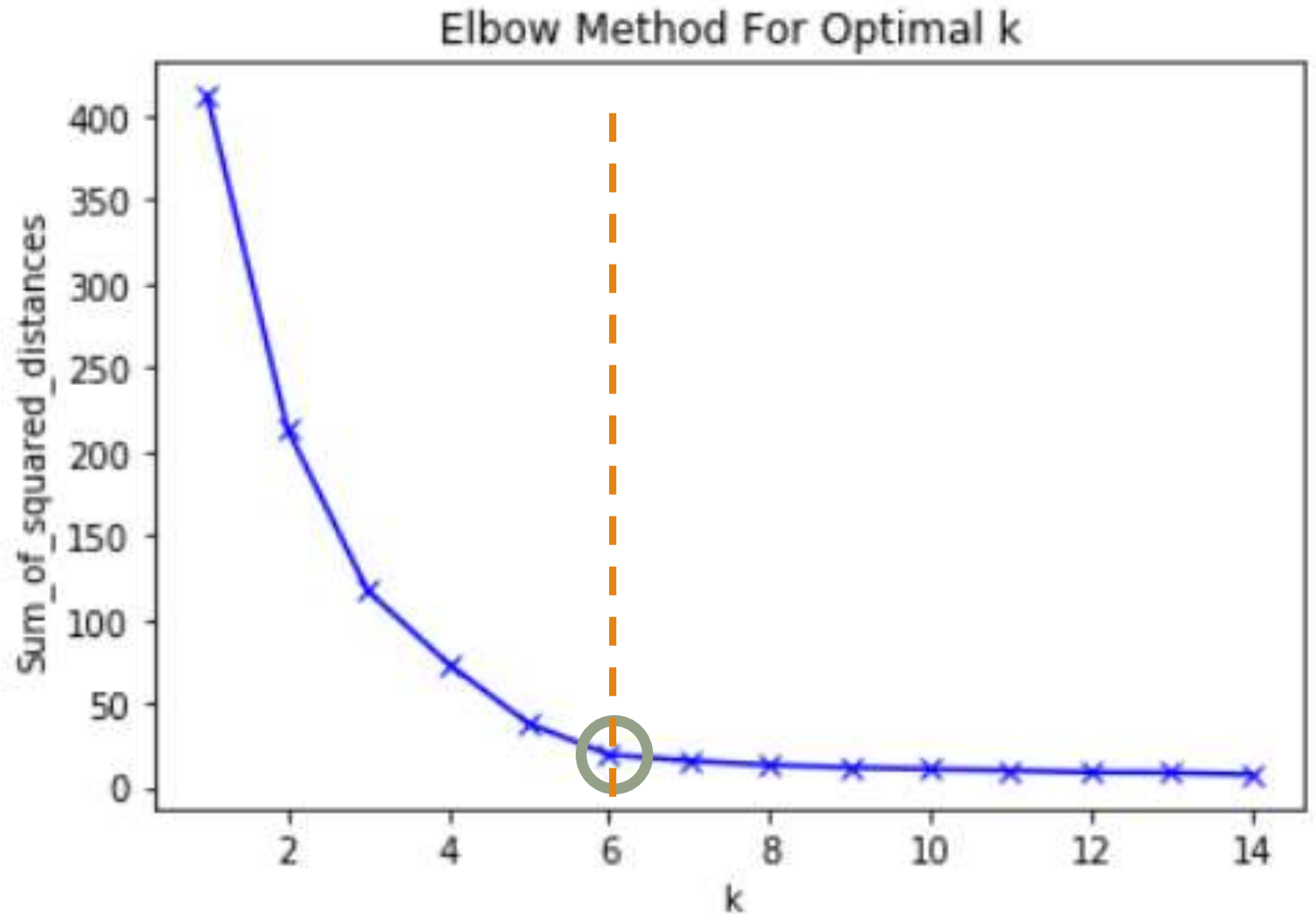


Algunas notas

- Altamente dependiente de la condición inicial
- Debemos “adivinar” el número de clúster: probar y evaluar
 - Demasiados y divides lo que debería ser un solo clúster
 - Muy pocos y combinas clúster, o separas y combinas otros.
- Computacionalmente es costoso $\mathcal{O}(kN)$

¿Cuántos Clústers?

- Método del codo



K-Means ++

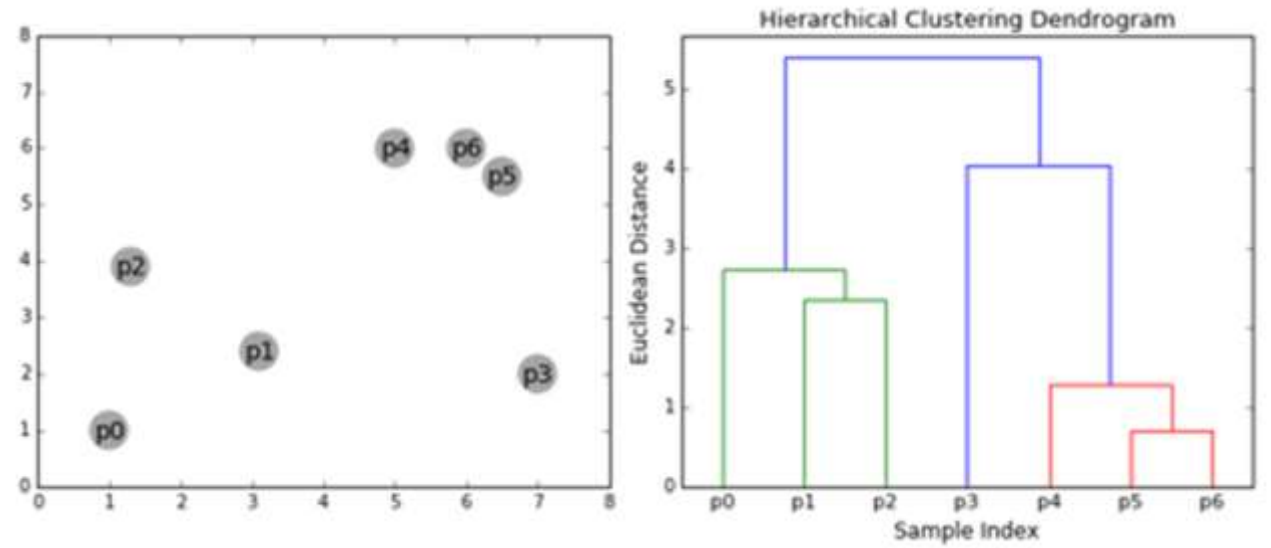
1. Seleccionar aleatoriamente un punto para que sea un centroide
2. Para cada punto calcular la distancia D entre el punto y el centroide más cercano ya creado
3. Elegir un nuevo punto para crear el siguiente centroide, usando una probabilidad ponderada. $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$
4. Repetir desde 2 hasta crear los K centroides

K-means ++

- Los centroides iniciales estarán más distribuidos de a lo largo de la data y a lo largo de los posibles clusters
- <http://shabal.in/visuals/kmeans/KMeansPlusPlus.pdf>

An orange brushstroke graphic on the left side of the slide, with a rough, hand-painted edge.

Demo con MNIST



Clustering Jerárquico

TAMBIÉN LLAMADO AGLOMERATIVO

Clustering Jerárquico

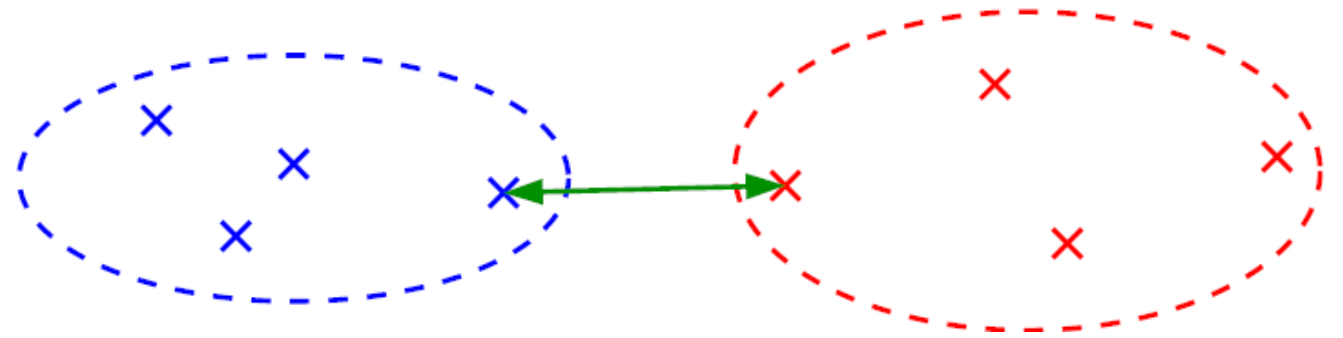
- Los clúster en k-means son “planos”, sin estructura
- Si ocurre un cambio en k, debemos recalcular
- Clustering jerárquico es de abajo hacia arriba
- Permite construir las relaciones
- Todos los grados del clúster pueden ser obtenidos.

Algoritmo

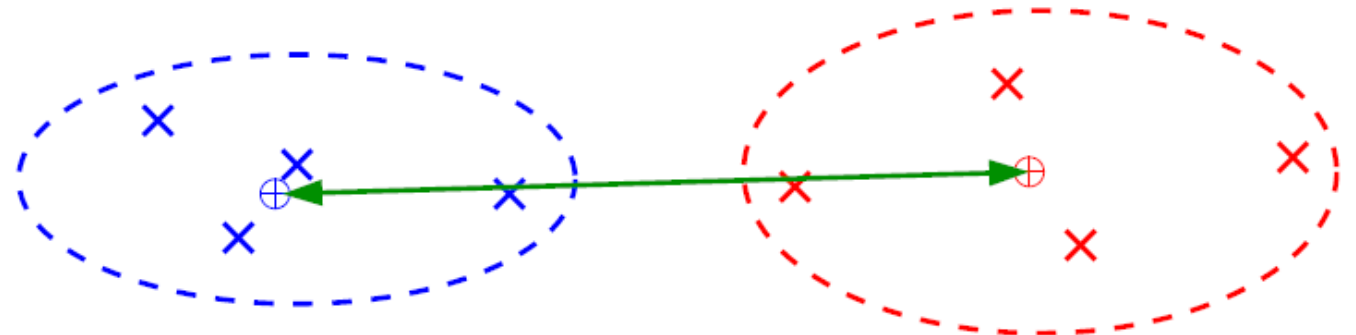
1. Computar la distancia entre todos los pares de puntos
2. Encontrar el par más cercano
3. Reemplazar ese par con el par agrupado
4. Recalcular la distancia entre todos los puntos y el nuevo par usando estrategias de enlace
5. Ir a 2, y continuar agrupando hasta que todos los puntos sean agrupados

Estrategias de enlace

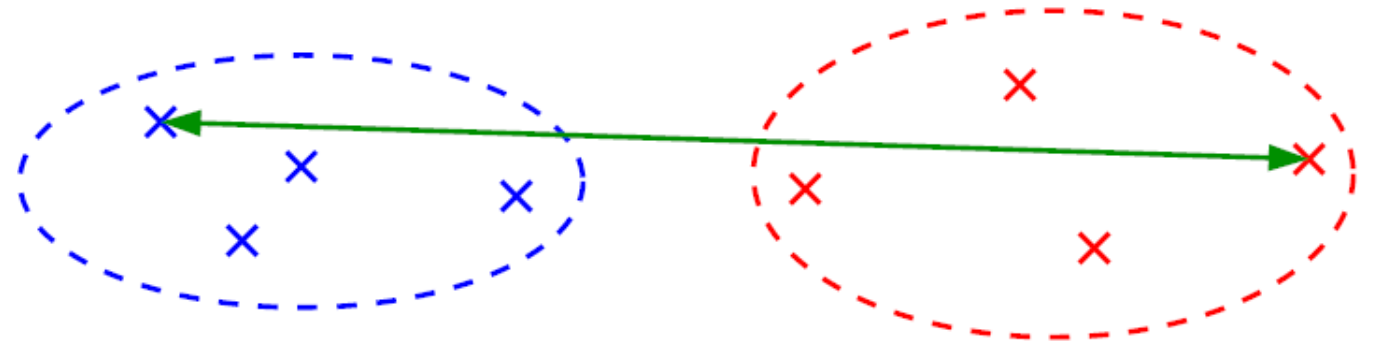
- Mínimo o simple



- Promedio



- Máximo o completo



Ejemplo

Lista de puntos

Agrupar 1 y 2

Agrupar 3 y 4

Agrupar 0 con (1,2)

Agrupar 5 y 6

Agrupar (3,4) y (0,(1,2))

Agrupar (5,6) y ((3,4),(0,(1,2)))

0,1,2,3,4,5,6

0,3,4,5,6,(1,2)

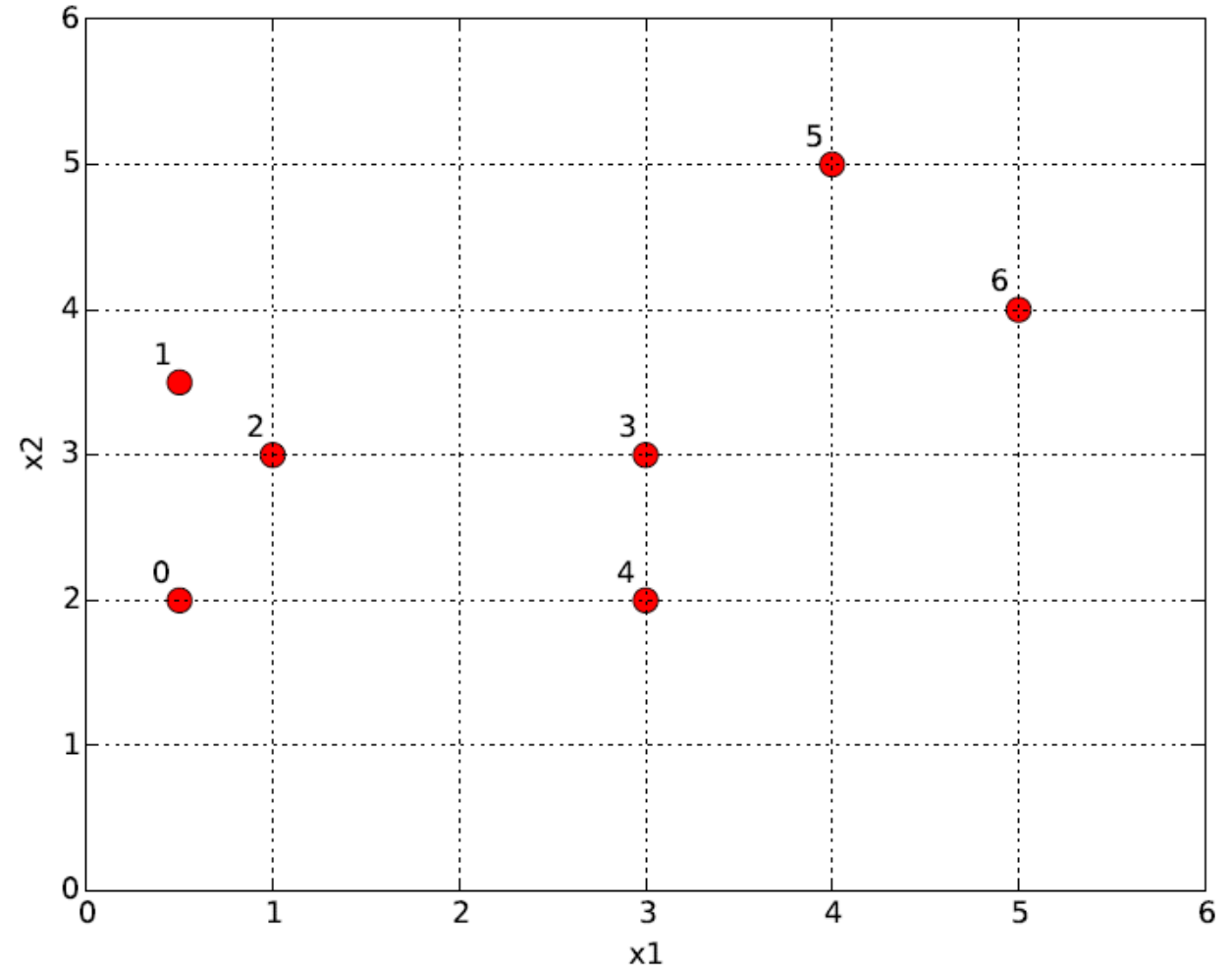
0,5,6,(1,2),(3,4)

5,6,(3,4),(0,(1,2))

(3,4),(0,(1,2)),(5,6)

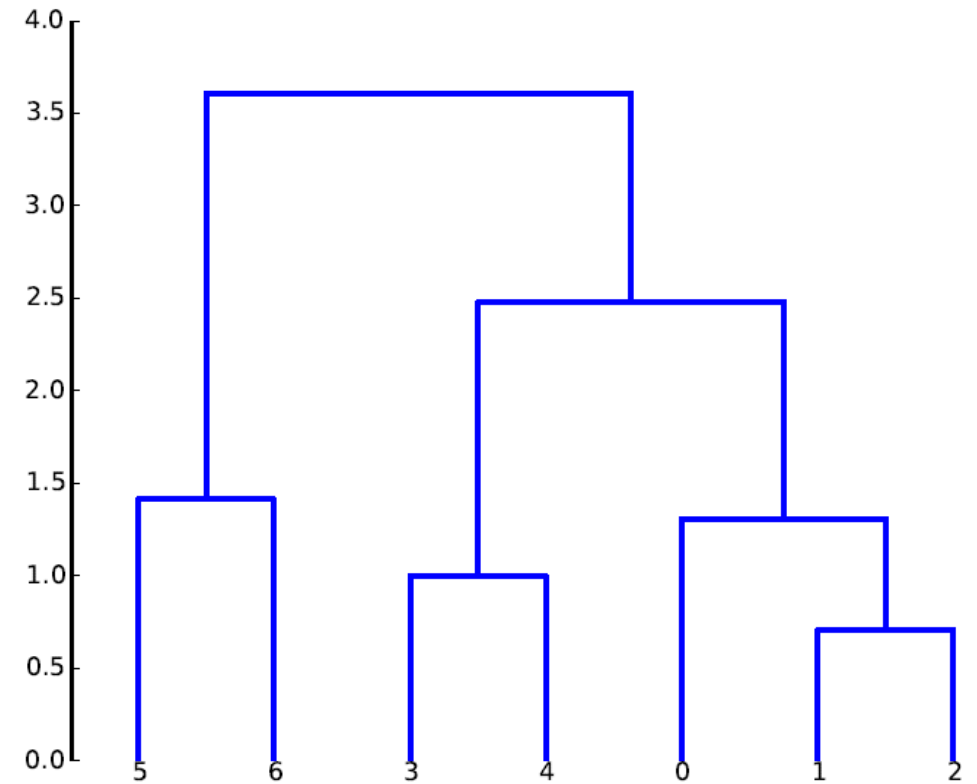
(5,6),((3,4),(0,(1,2)))

((5,6),((3,4),(0,(1,2))))



Dendograma

- Representación gráfica de la relación de los puntos.
- La altura representa la distancia



Notas sobre clustering jerárquico

- Para determinar un grupo, cortar horizontalmente a lo largo del dendograma
- Computacionalmente costoso $\mathcal{O}(N^2)$
- K-means es más rápido si se sabe el número de clúster a buscar
- Normalmente no dan el mismo resultado