Aprendizaje No Supervisado

Machine Learning

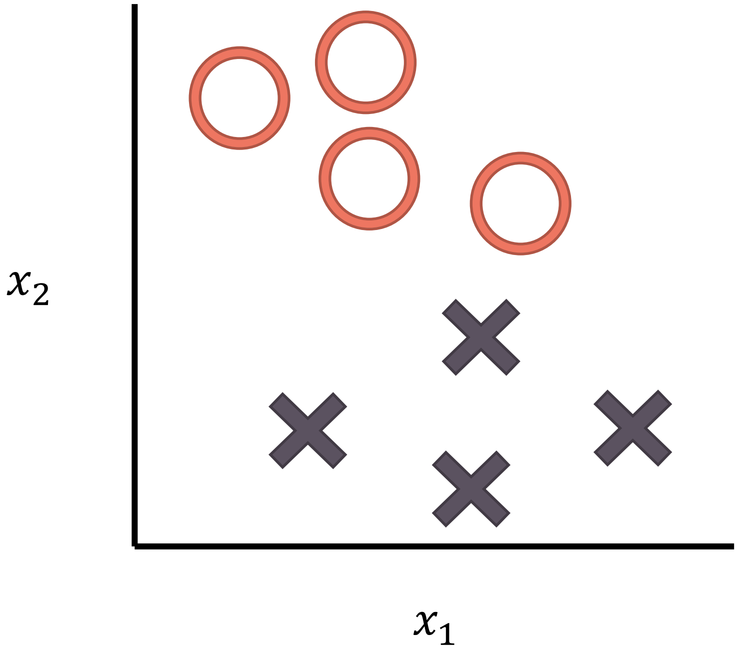
Agenda

1. ¿Qué es el Aprendizaje No Supervisado?
2. K-Means
   1. Idea General
   2. Algoritmo de K-Means
   3. Inicialización de K-Means
   4. ¿Cómo elegir K?
3. K-Means++

¿Qué es el Aprendizaje No Supervisado?

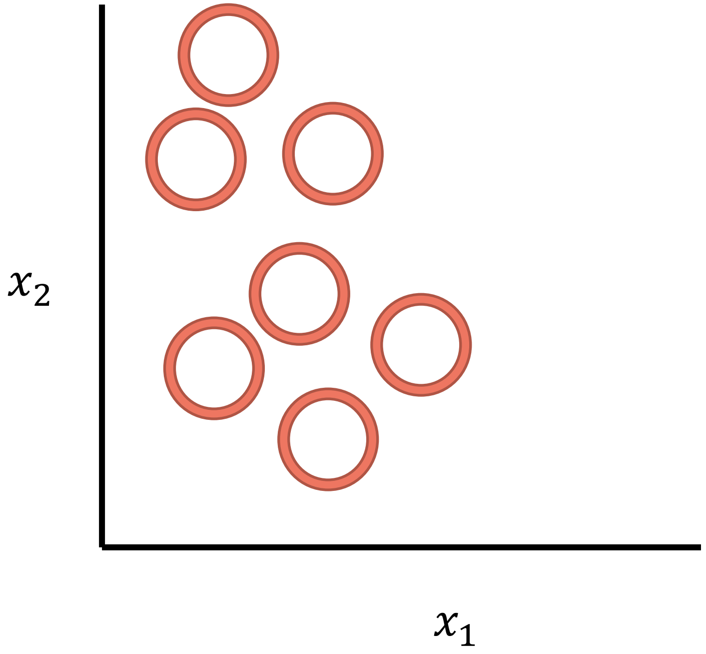
Aprendizaje No Supervisado

¡Debemos encontrar patrones en los datos!



Aprendizaje Supervisado

$$\{(x^{(1)}, y^{(1)}), \dots, (x^{(n)}, y^{(n)})\}$$



Aprendizaje No Supervisado

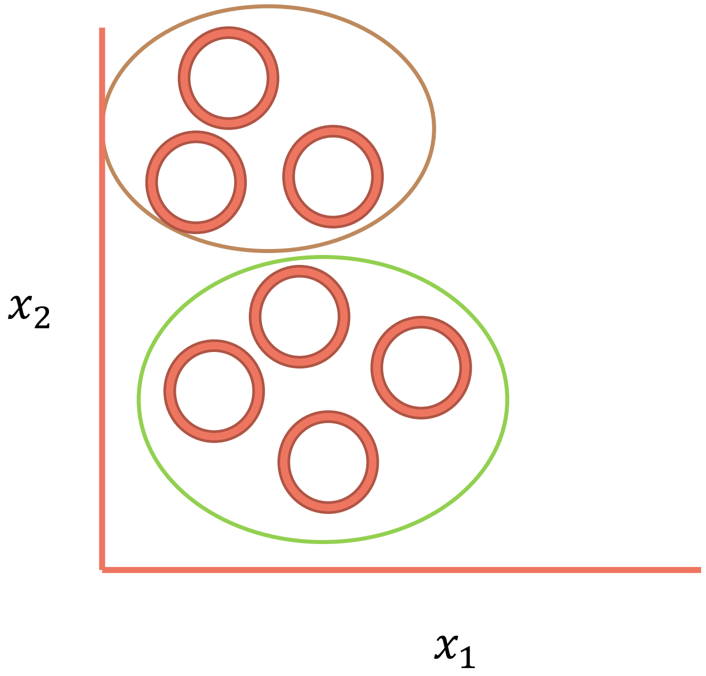
$$\{x^{(1)}, x^{(2)}, \dots, x^{(n)}\}$$

Aprendizaje No Supervisado

- En el aprendizaje no supervisado no se dan las clases o valores correctos de los datos.
- ¿Por qué? No siempre es posible determinar el número de clases de antemano, o es caro o difícil determinarlas.
- Aquí la tarea es encontrar estructuras o patrones en los datos.

Aprendizaje No Supervisado

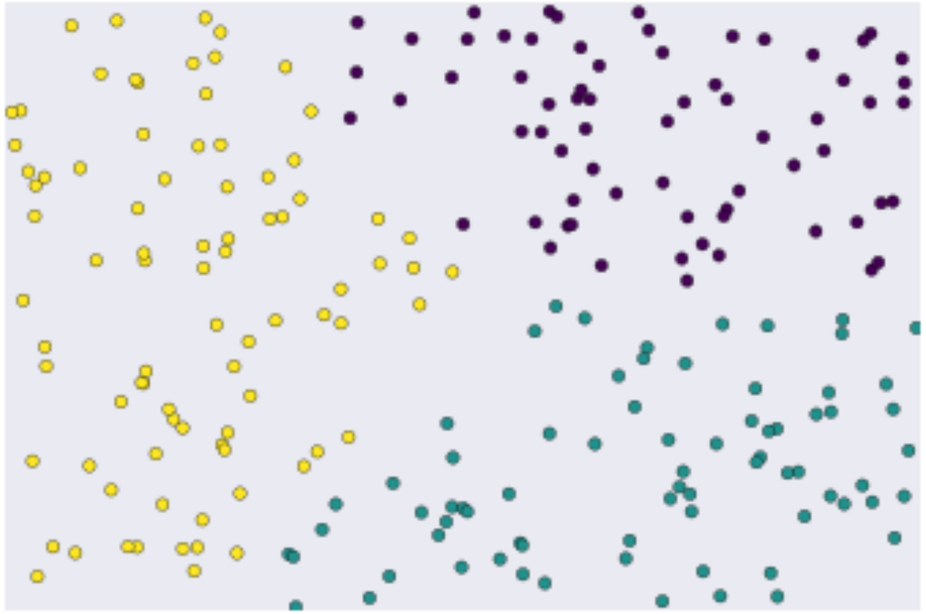
- En el aprendizaje no supervisado no se dan las clases o valores correctos de los datos.
- ¿Por qué? No siempre es posible determinar el número de clases de antemano, o es caro o difícil determinarlas.
- Aquí la tarea es encontrar estructuras o patrones en los datos.
- Se busca detectar clústeres en los datos.



¿Qué es un clúster?

Esta foto de Autor desconocido está bajo licencia CC BY-NC-ND

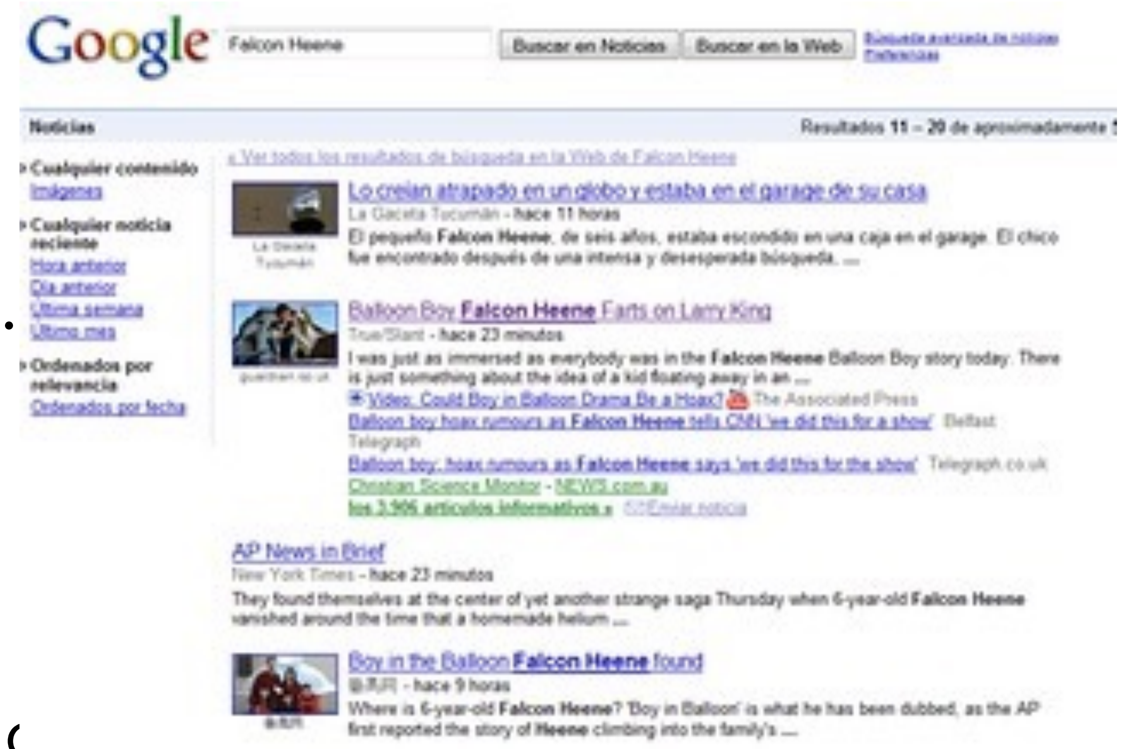
Año	(Década del 2010)
Género	(Alternativo)
Lugar	(Estados Unidos)
Ambiente Social	(Empoderamiento)



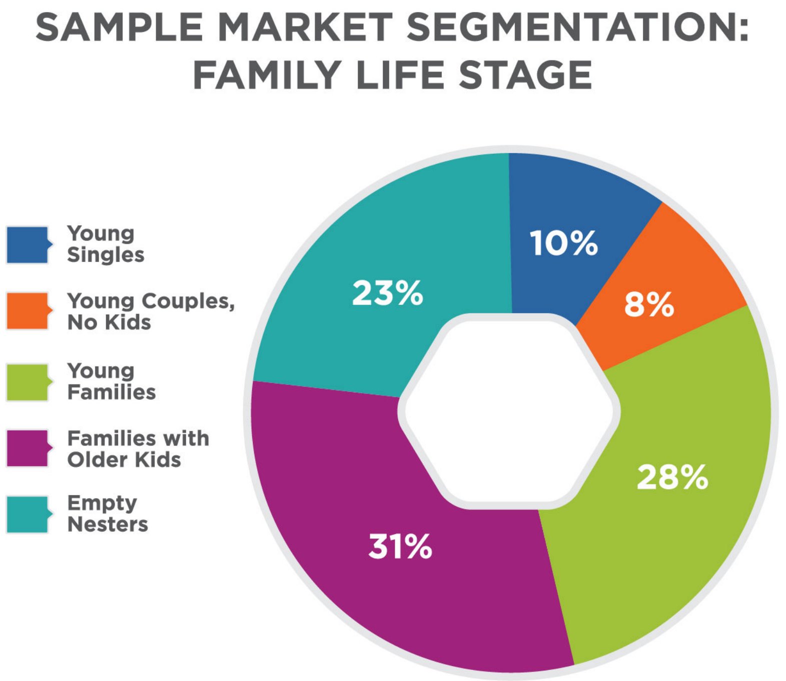
Aprendizaje No Supervisado

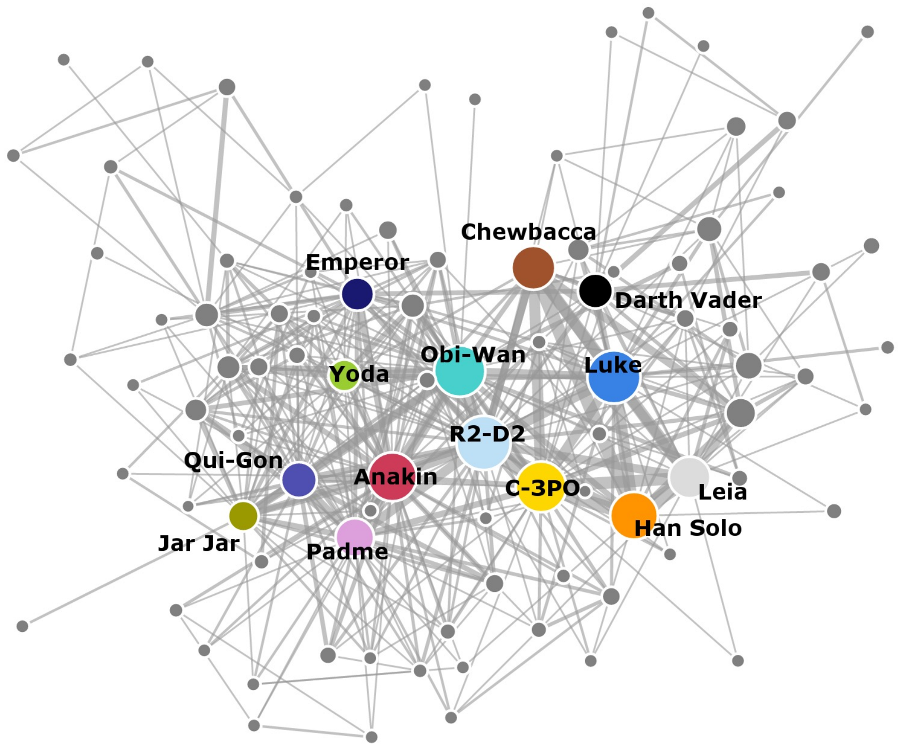
Una tarea común es en la clasificación de noticias:

- No sabemos en cuantas clases separar cada noticias. E.g., deportes, sociales, nacional, internacional, etc.
- Una forma es determinar clústeres por medio de similitud en cuanto a temas c palabras.



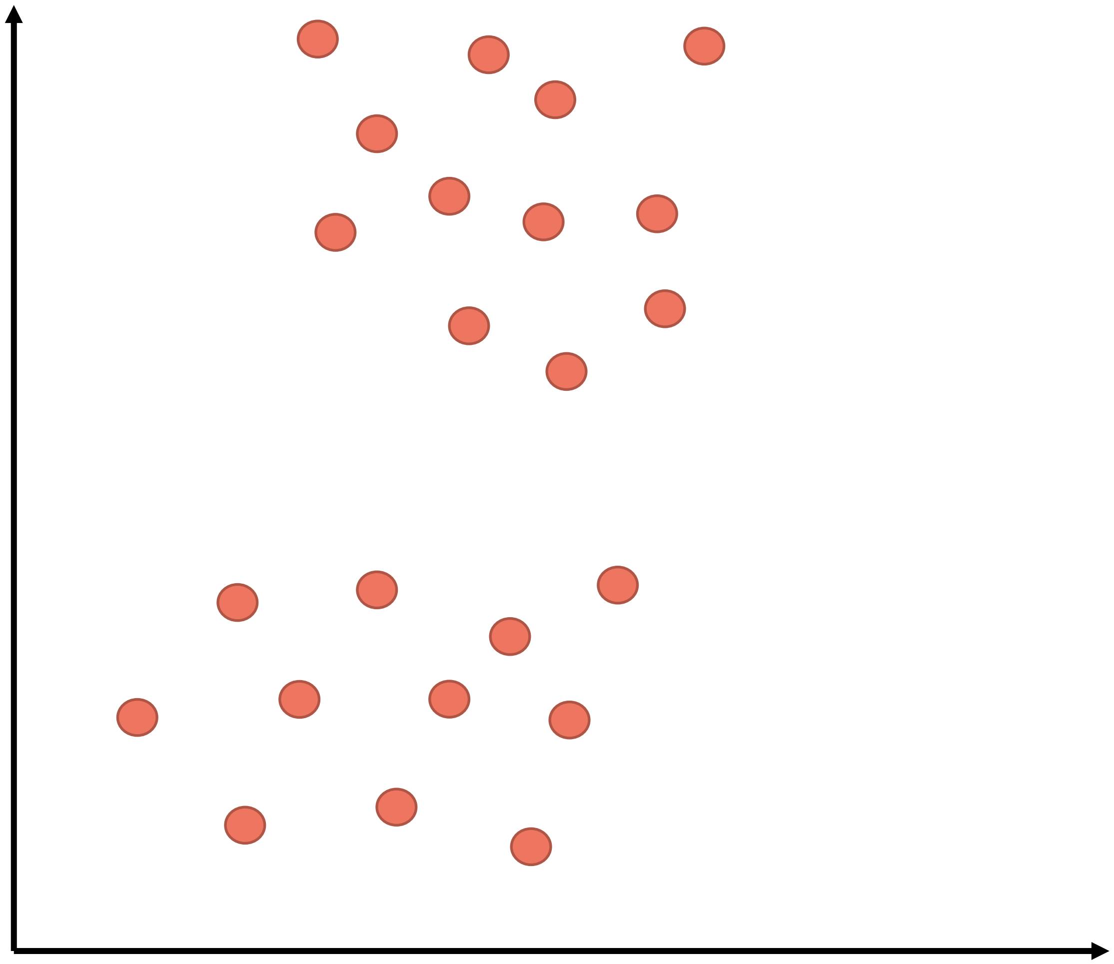
Aprendizaje No Supervisado





K-Means

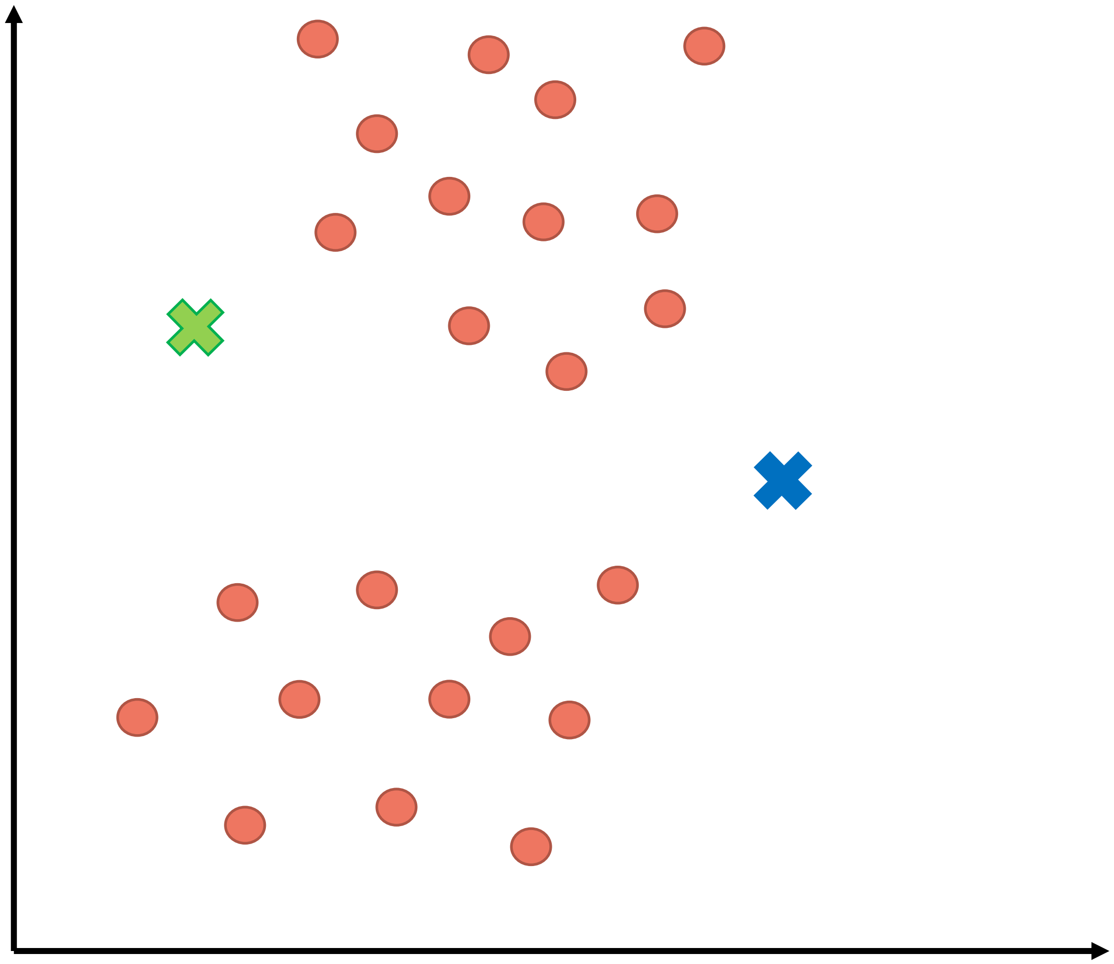
K-Means



Tenemos nuestros datos, que por cuestiones de explicación, **solo cuentan con dos dimensiones** x_1 y x_2.

Además, deseamos **dividir los datos en 2 grupos distintos**, es decir, 2 clústeres.

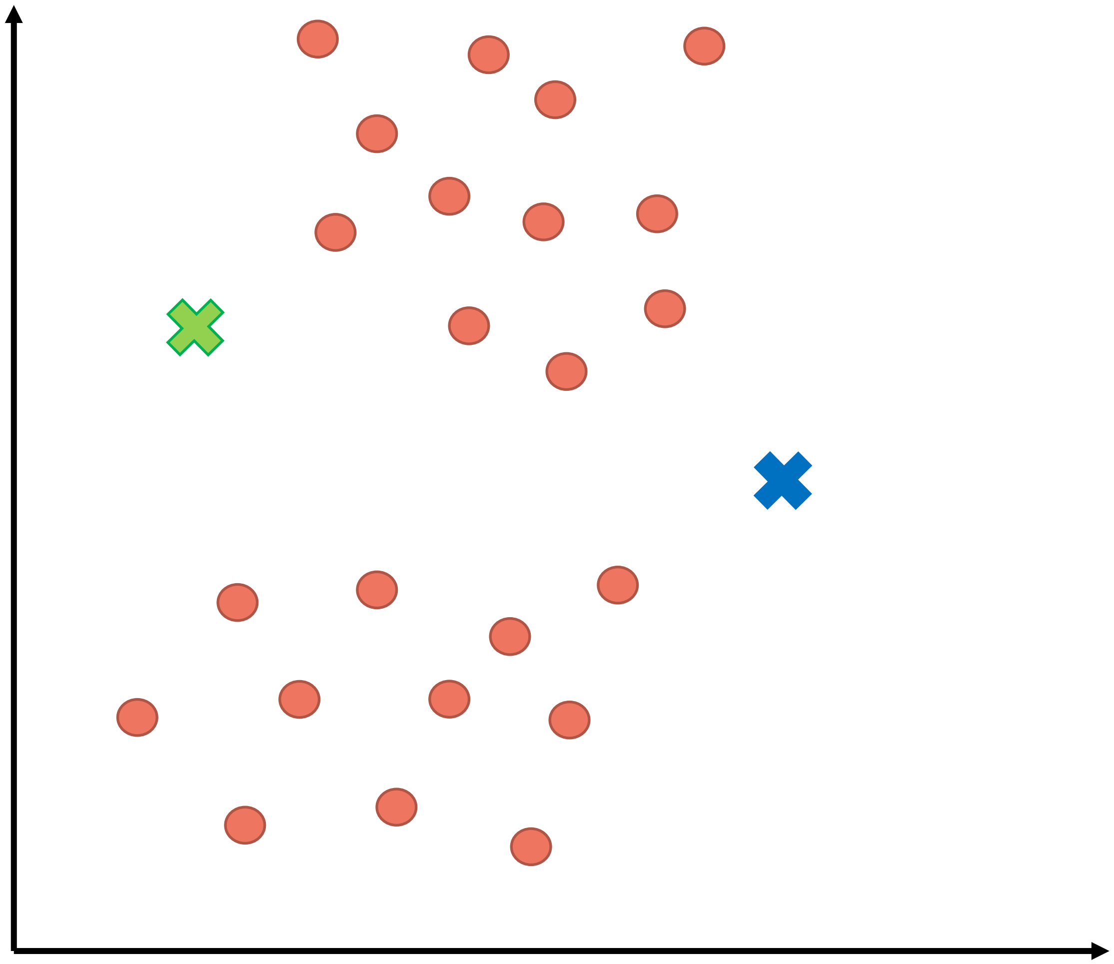
K-Means



Iniciamos el algoritmo **proponiendo dos centroides**, uno por cada clúster que se quiere encontrar. Usualmente es al azar.

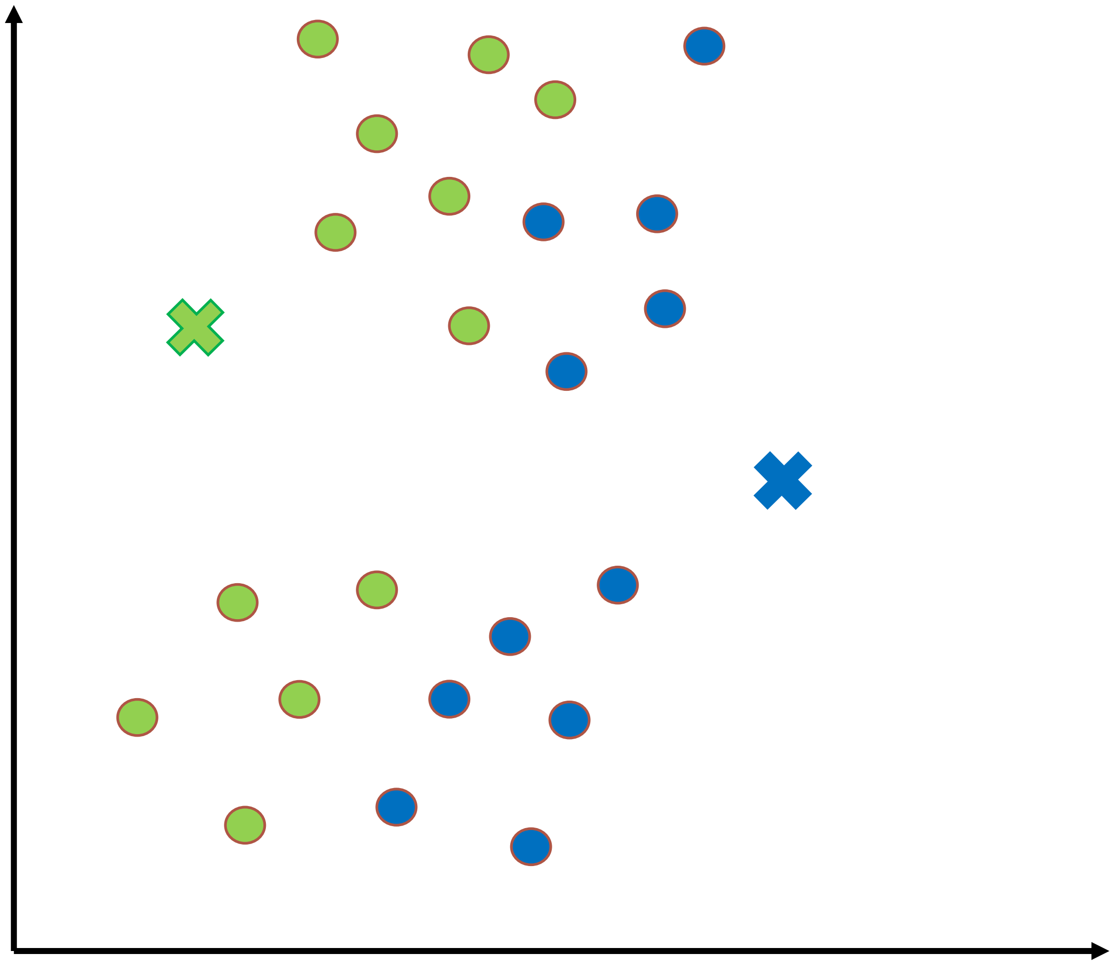
Estos puntos se conocen como **los centroides del clúster**.

K-Means



El primer paso consiste en **asignar cada punto a uno de los centroides** según su distancia. Es decir, **se asigna al más cercano**.

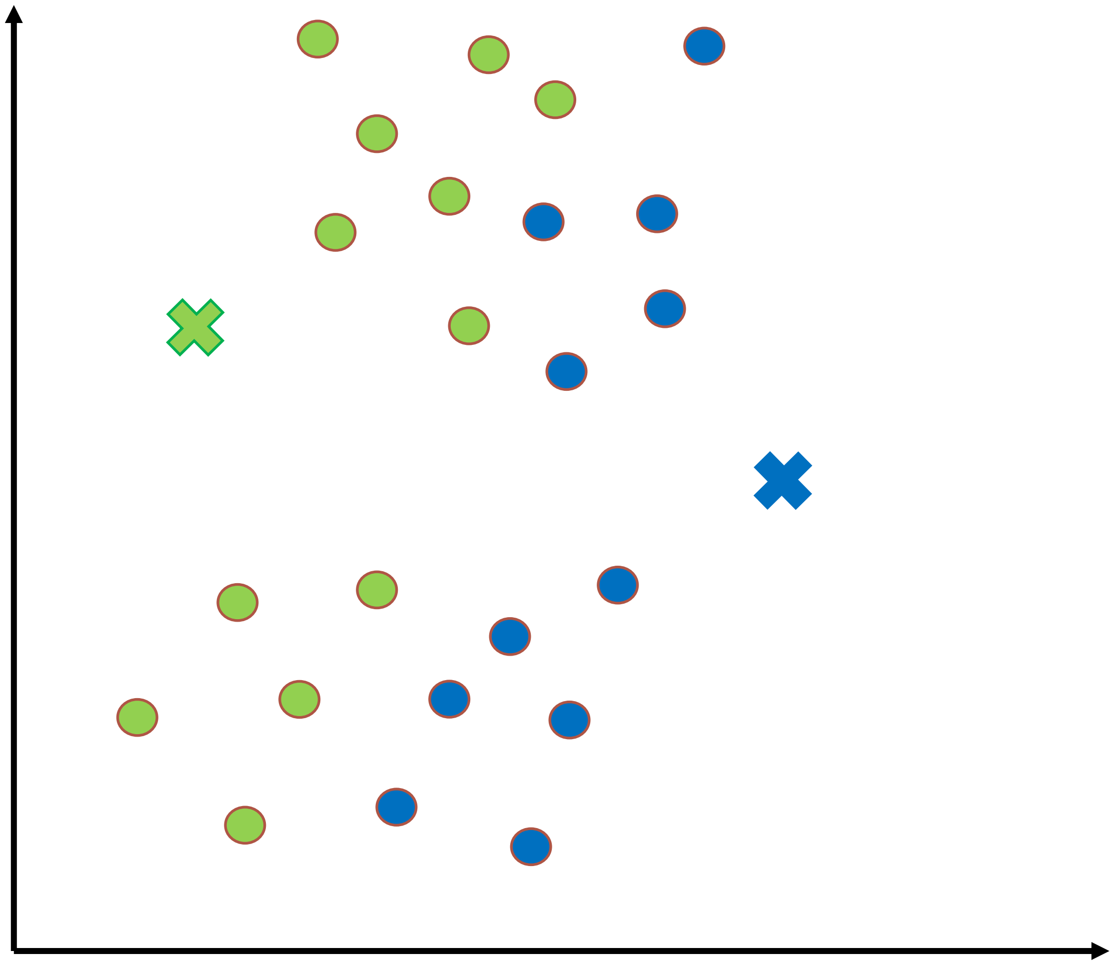
K-Means



El primer paso consiste en **asignar cada punto a uno de los centroides** según su distancia. Es decir, **se asigna al más cercano**.

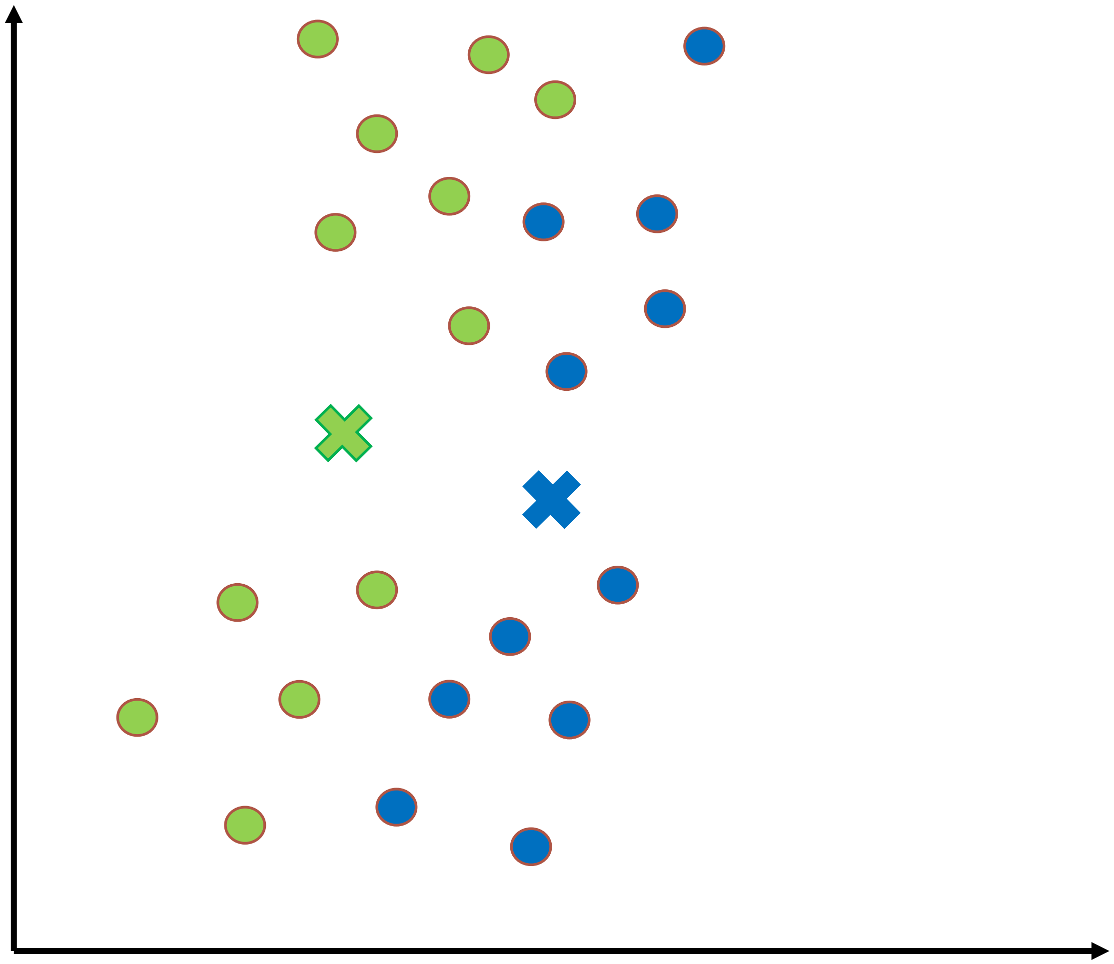
Este paso se le conoce como **asignación del clúster**.

K-Means



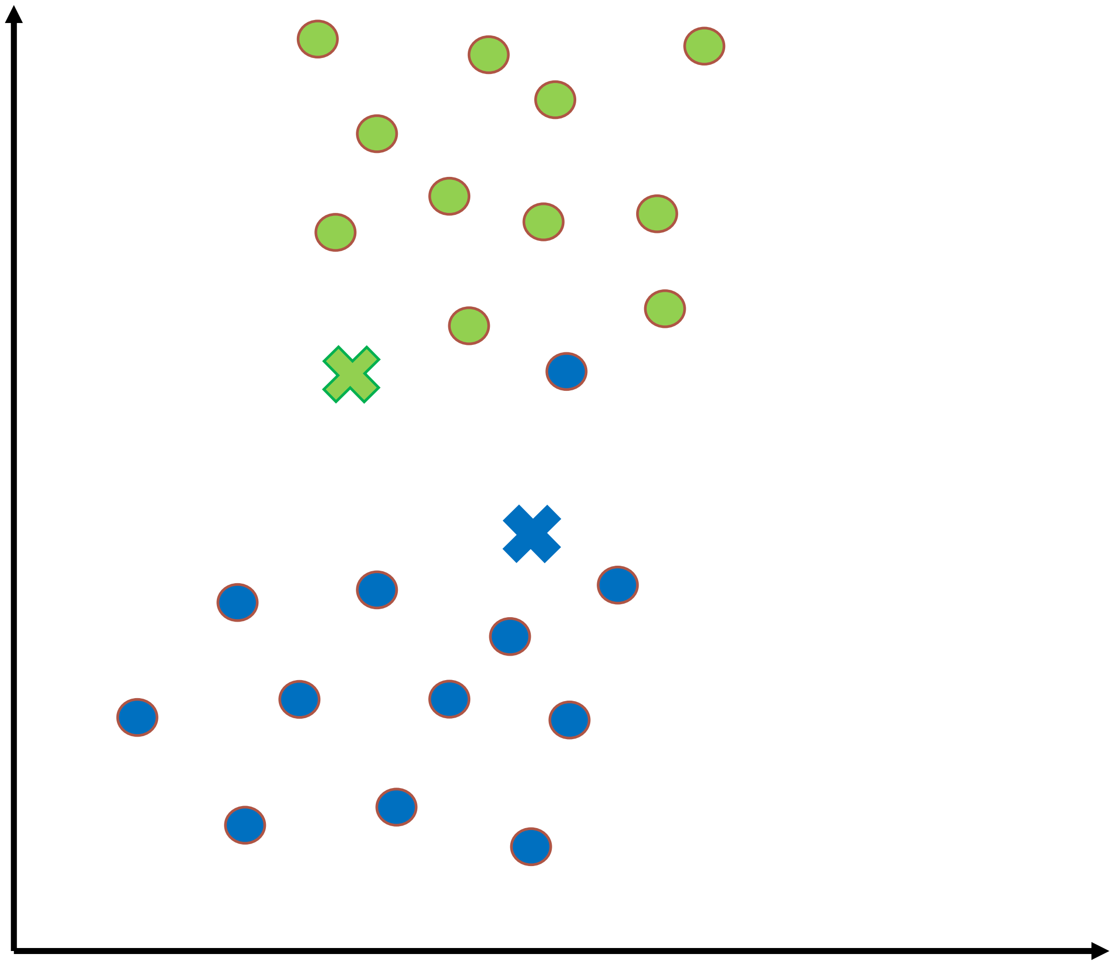
El segundo paso consiste en **mover los centroides**. Lo que sucede aquí es mover la ubicación de los centroides **al promedio o media de los puntos del mismo color**.

K-Means



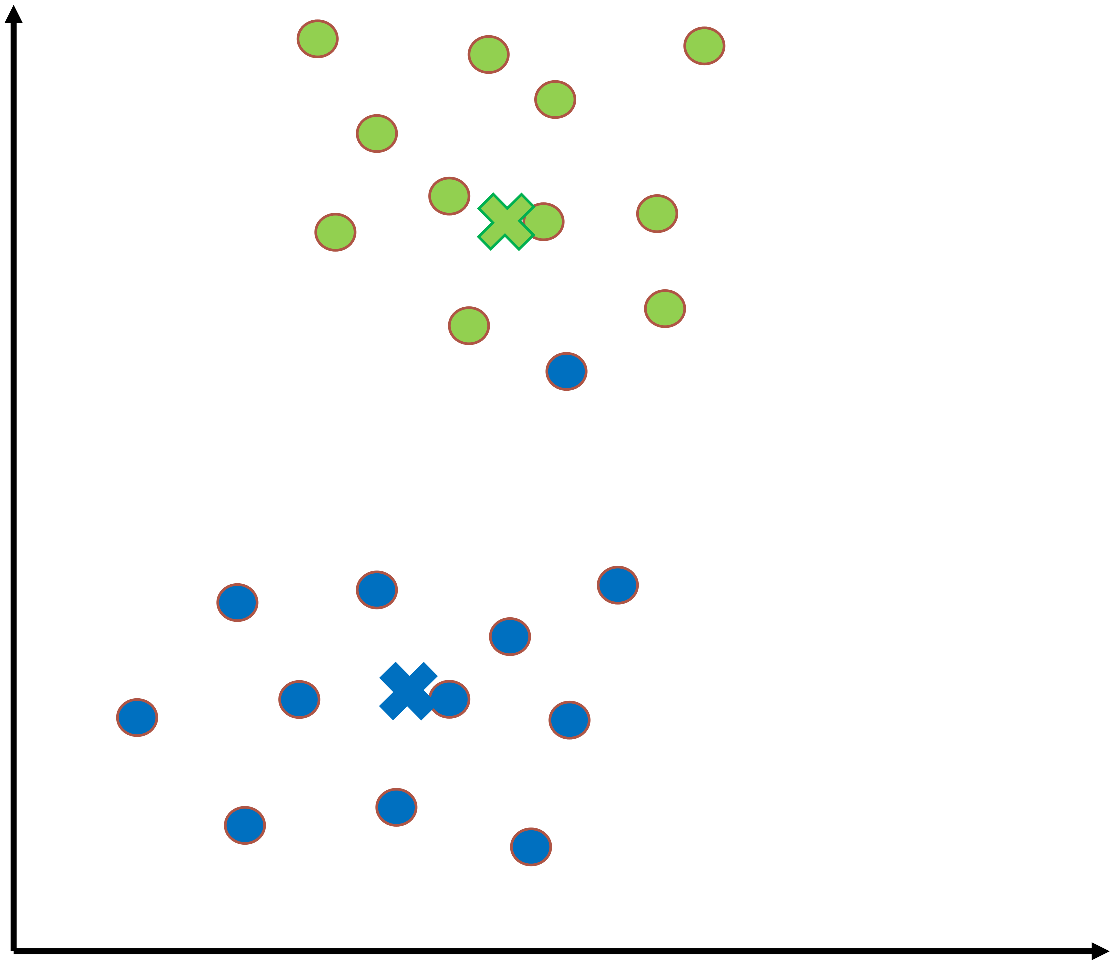
El segundo paso consiste en **mover los centroides**. Lo que sucede aquí es mover la ubicación de los centroides **al promedio o media de los puntos del mismo color**.

K-Means



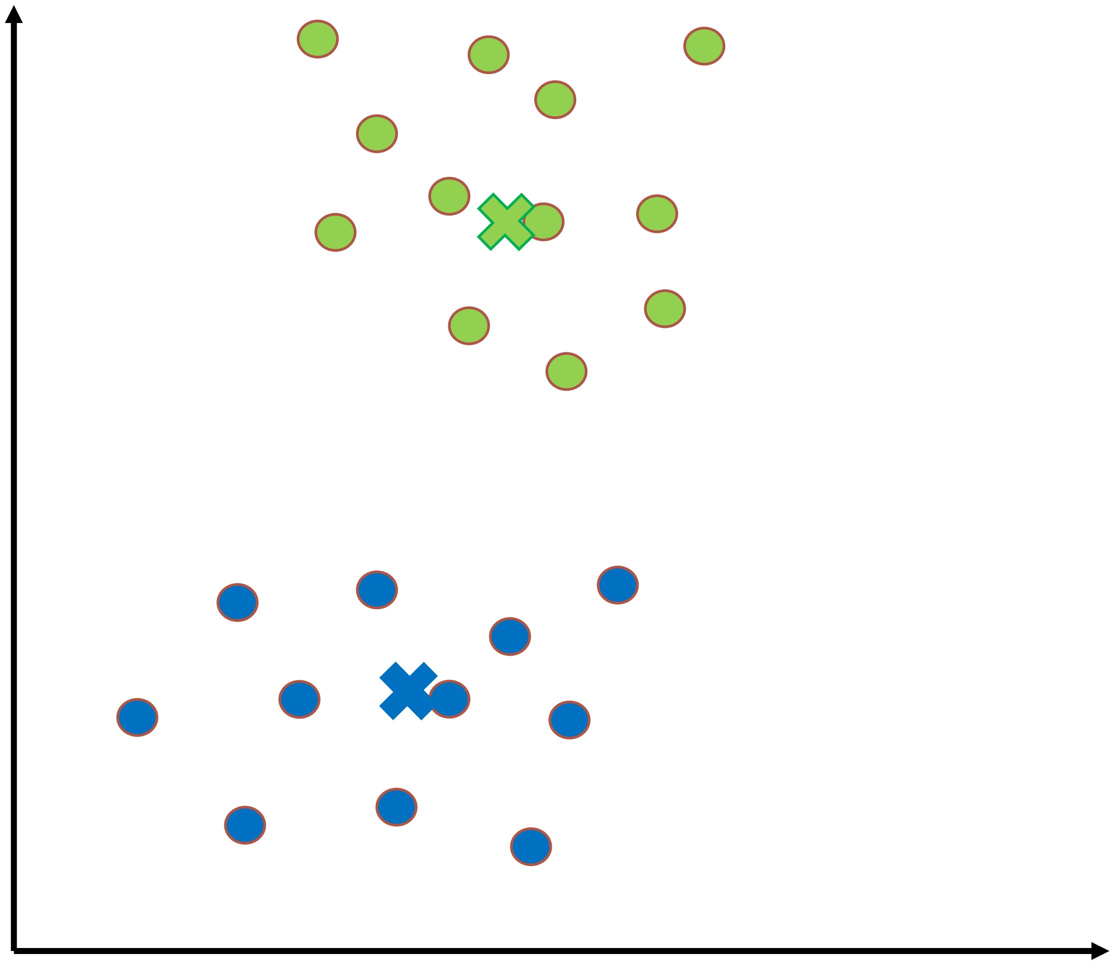
Se **repite el paso de asignación** de clúster con las nuevas ubicaciones.

K-Means



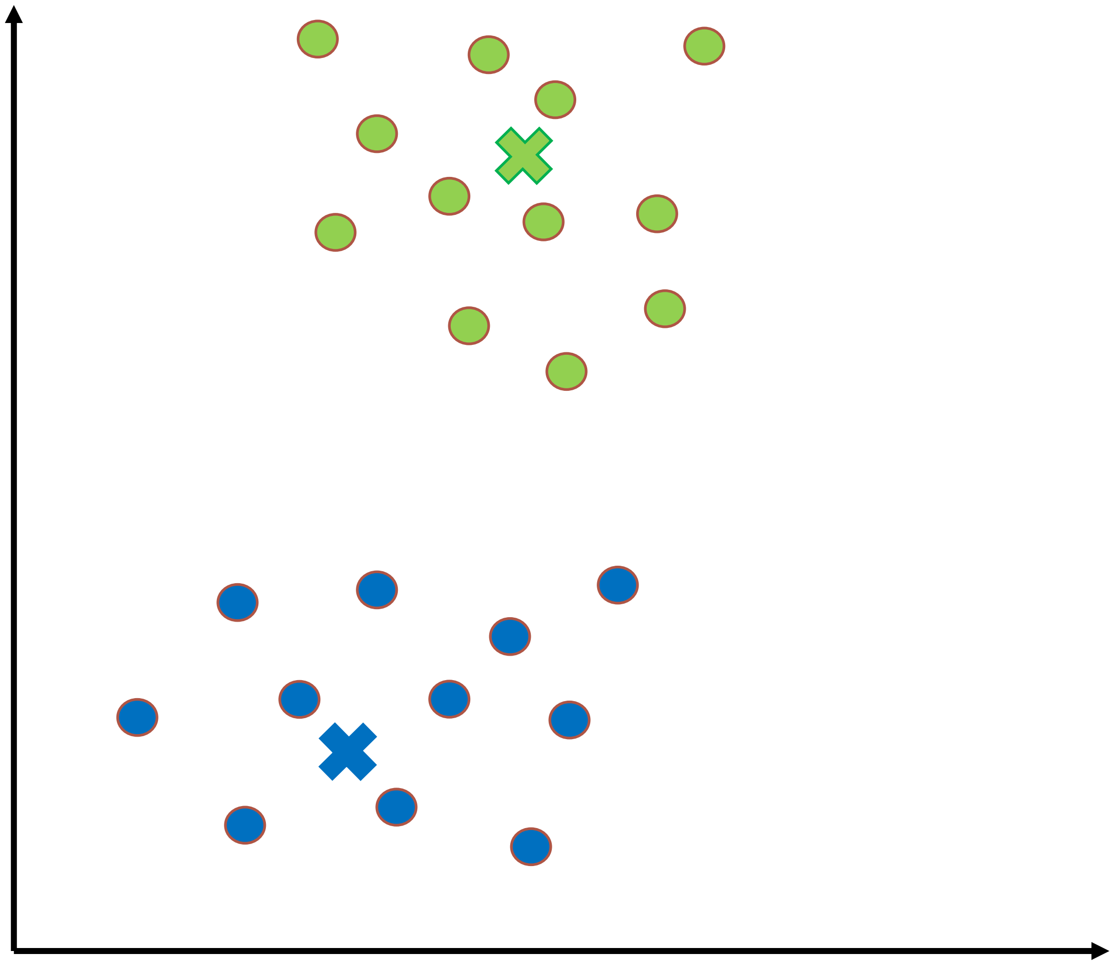
Se **repite el paso de movimiento de los centroides** del clúster con los nuevos puntos.

K-Means



Se **repite el paso de asignación** de clúster con las nuevas ubicaciones.

K-Means



Se **repite el paso de movimiento de los centroides** del clúster con los nuevos puntos.

Es posible **repetir el proceso de asignación y movimiento**, pero de aquí en adelante daría los mismos resultados.

Algoritmo de K-Means

Nosotros debemos proponer el valor de K.

El conjunto de datos sin etiquetas.

Entrada:

- K (número de clústeres)
- Conjunto de entrenamiento $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$
- $x^{(i)} \in \mathbb{R}^m$

Es un vector de m características.

Algoritmo de K-Means

1. Inicializar aleatoriamente los K centroides de los clústeres $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^m$.
2. Repetir:
   a) Para $i = 1$ hasta n:
      $c^{(i)} :=$ índice del centroide del clúster más cercano a $x^{(i)}$
   b) Para k $= 1$ hasta K:
      $\mu_k :=$ media o promedio de los puntos asignados al clúster k.

Algoritmo de K-Means

1. Inicializar aleatoriamente los K centroides de los clústeres $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^m$.
2. Repetir:

$$\boxed{c^{(i)} = \min_k \left\| x^{(i)} - \mu_k \right\|^2}$$

Asignación

   a) Para $i = 1$ hasta n:

      $c^{(i)} :=$ índice del centroide del clúster más cercano a $x^{(i)}$

   b) Para k = 1 hasta K:

      $\mu_k :=$ media o promedio de los puntos asignados al clúster k.

Algoritmo de K-Means

1. Inicializar aleatoriamente los K centroides de los clústeres $\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^m$.
2. Repetir:

$$c^{(i)} = \min_k \left\| x^{(i)} - \mu_k \right\|^2$$

Asignación

   a) Para $i = 1$ hasta n:

   $c^{(i)} :=$ índice del centroide del clúster más cercano a $x^{(i)}$

Movimiento

   b) Para k = 1 hasta K:

   $\mu_k :=$ media o promedio de los puntos asignados al clúster k.

$$x^{(1)}, x^{(5)}, x^{(7)}, x^{(12)}$$
$$\downarrow$$
$$c^{(1)} = 2, c^{(5)} = 2, c^{(7)} = 2, c^{(12)} = 2$$

$$\mu_2 = \frac{1}{4}\left[x^{(1)} + x^{(5)} + x^{(7)} + x^{(12)}\right] \in \mathbb{R}^m$$

Algoritmo de K-Means

$c^{(i)}$ = índice del clúster (1,…,K) que se asigna al vector de datos $x^{(i)}$ en la actual iteración
μ_k = centroide del clúster $k, \mu_k \in \mathbb{R}^m$
$\mu_c^{(i)}$ = centroide del clúster que se asigna al vector $x^{(i)}$

$$x^{(i)} \rightarrow 5, c^{(i)} \rightarrow 5, \mu_c^{(i)} \rightarrow \mu_5$$

Función Objetivo (*Distortion Cost Function*):

$$J\left(c^{(1)}, \dots, c^{(n)}; \mu_1, \dots, \mu_K\right) = \frac{1}{n}\sum_{i=1}^{n} \left\| x^{(i)} - \mu_c^{(i)} \right\|^2$$

$$\min_{c^{(i)},\dots,c^{(n)},\mu_i,\dots,\mu_K} J\left(c^{(1)}, \dots, c^{(n)}; \mu_1, \dots, \mu_K\right)$$

Algoritmo de K-Means

1. Inicializar aleatoriamente los K centroides de los clústeres $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^m$.
2. Repetir:
   a) Para $i = 1$ hasta n:

      $$\min_{c^{(i)},\ldots,c^{(n)}} J\left(c^{(1)}, \ldots, c^{(n)}; \mu_1, \ldots, \mu_K\right)$$

      $c^{(i)} :=$ índice del centroide del clúster más cercano a $x^{(i)}$
   b) Para k $= 1$ hasta K:

      $\mu_k :=$ media o premedio de los puntos asignados al clúster k.

      $$\min_{\mu_i,\ldots,\mu_K} J\left(c^{(1)}, \ldots, c^{(n)}; \mu_1, \ldots, \mu_K\right)$$

Algoritmo de K-Means

Supongamos que ejecutamos el algoritmo de K-Means y al finalizar obtenemos que:

$$c^{(1)} = 3, c^{(2)} = 5, c^{(3)} = 3, \dots$$

¿Cuáles de los siguientes enunciados son correctos?

1. El tercer vector de datos $x^{(3)}$ le fue asignado el clúster 3.
2. El primer y tercer vector de datos $x^{(1)}$ y $x^{(3)}$ fueron asignados al mismo clúster.
3. El segundo y tercer vector de datos $x^{(2)}$ y $x^{(3)}$ fueron asignados al mismo clúster.
4. De todos los posibles valores de $k \in \{1,2,\dots,K\}$, $k = 3$ minimiza la distancia entre $x^{(2)}$ y μ_k.

Algoritmo de K-Means -- Inicialización

Para $K = 2$

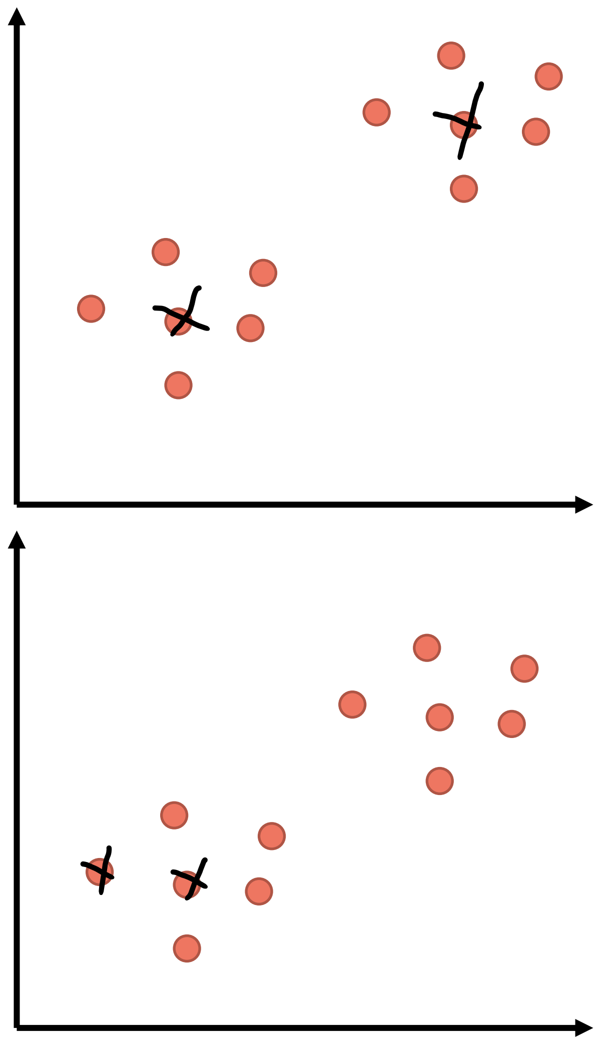
Algunos detalles prácticos:

1. $K < n$

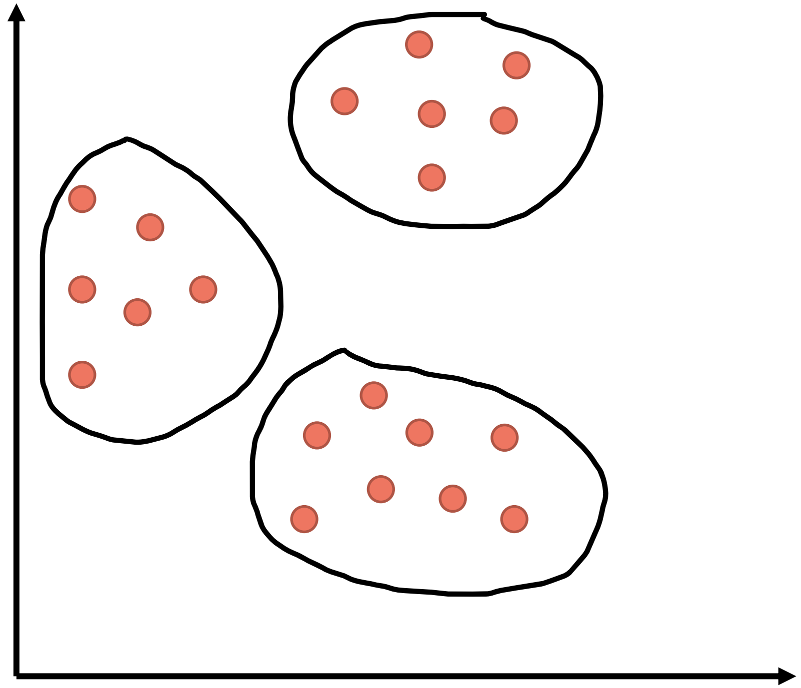
2. Elegir K puntos al azar del conjunto de datos.

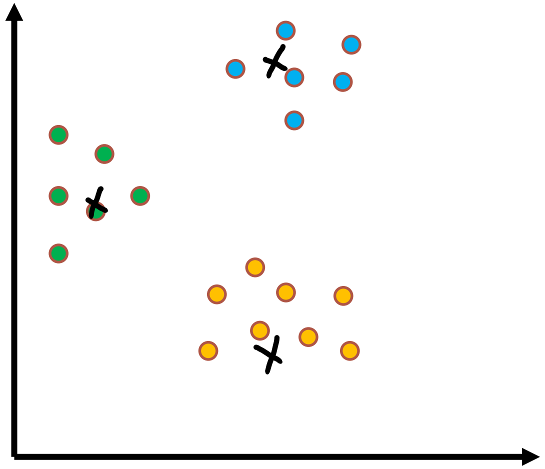
3. Asignar a $\mu_1, \dots, \mu_K$ los puntos elegidos al azar.

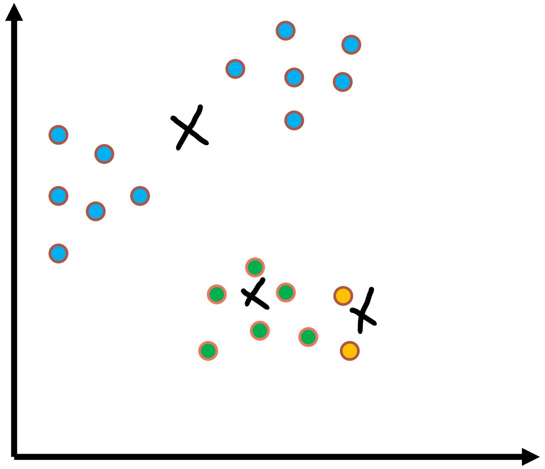
$$\mu_1 = x^{(i)}$$
$$\mu_2 = x^{(j)}$$
$$\vdots$$

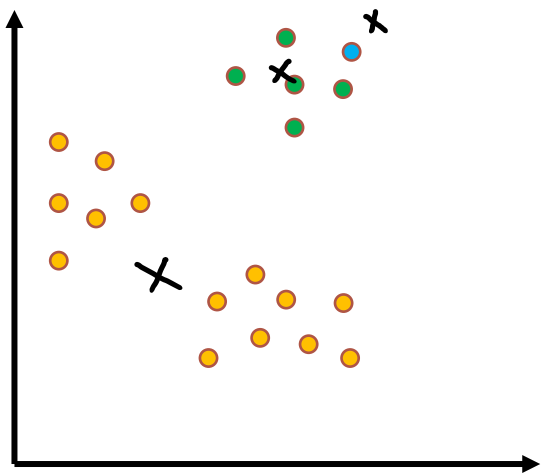


Algoritmo de K-Means -- Inicialización









Algoritmo de K-Means -- Inicialización

Primera opción – Repetir K-Means de tal manera que se elija aquel que tenga el menor valor en la función de costo:

Para $i = 1, \dots, 100$:

Inicializar aleatoriamente K-means.

Correr K-Means y obtener $c^{(1)}, \dots, c^{(n)}, \mu_1, \dots, \mu_K$

Calcular la función de costo (distorsión)

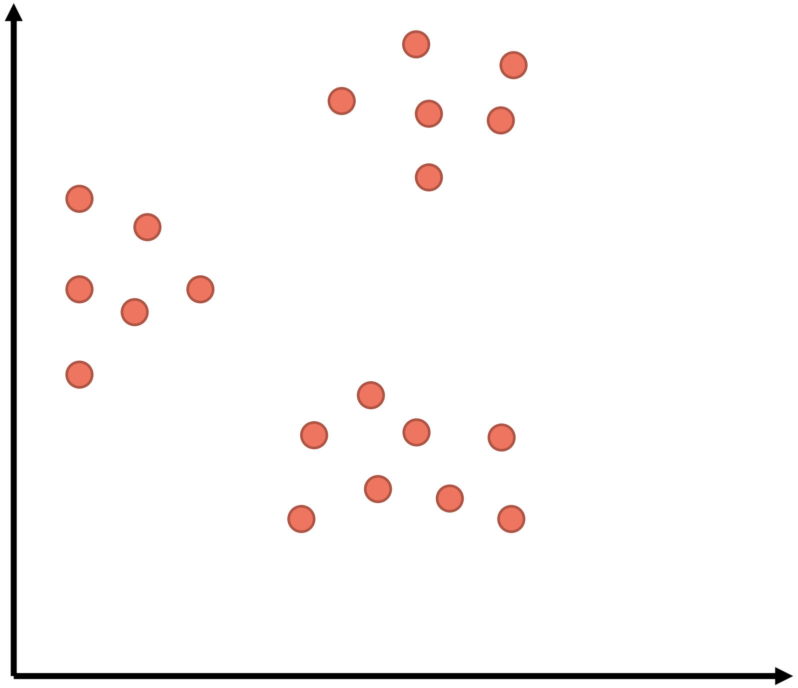
$$J(c^{(1)}, \dots, c^{(n)}, \mu_1, \dots, \mu_K)$$

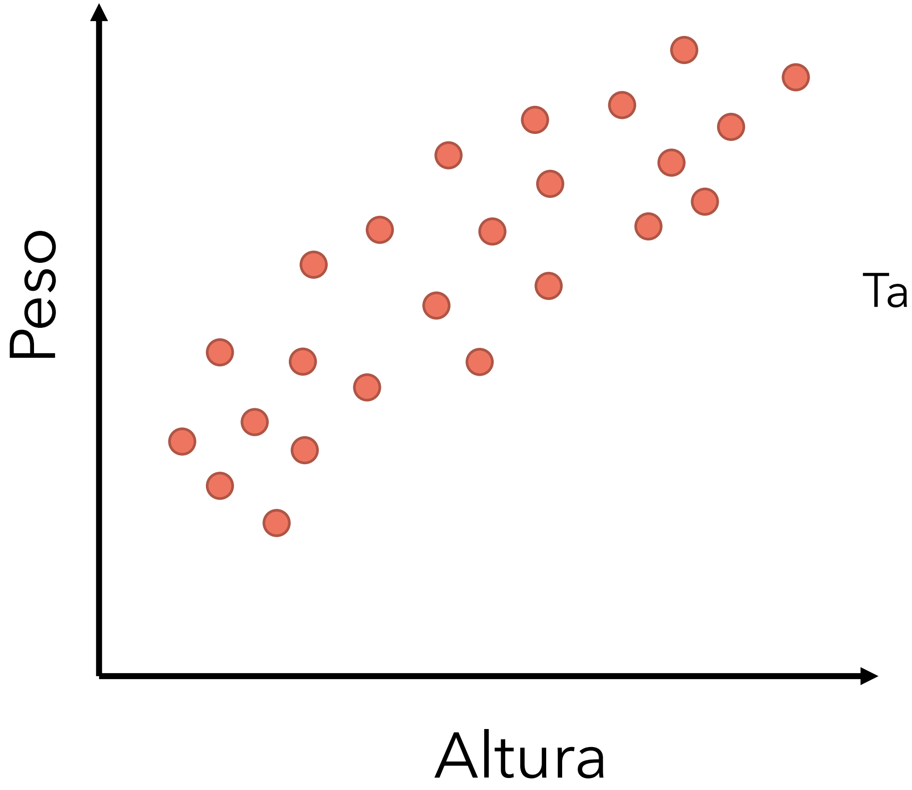
Elegir el clúster que tenga el menor valor de $J(c^{(1)}, \dots, c^{(n)}, \mu_1, \dots, \mu_K)$.

Algoritmo de K-Means

- ¿Qué pasa si un centroide no tiene puntos?
  - Se puede eliminar, de tal manera que queden $K - 1$ clústeres, o se puede reiniciar su ubicación al azar.
  - En práctica es muy difícil que llegue a suceder.

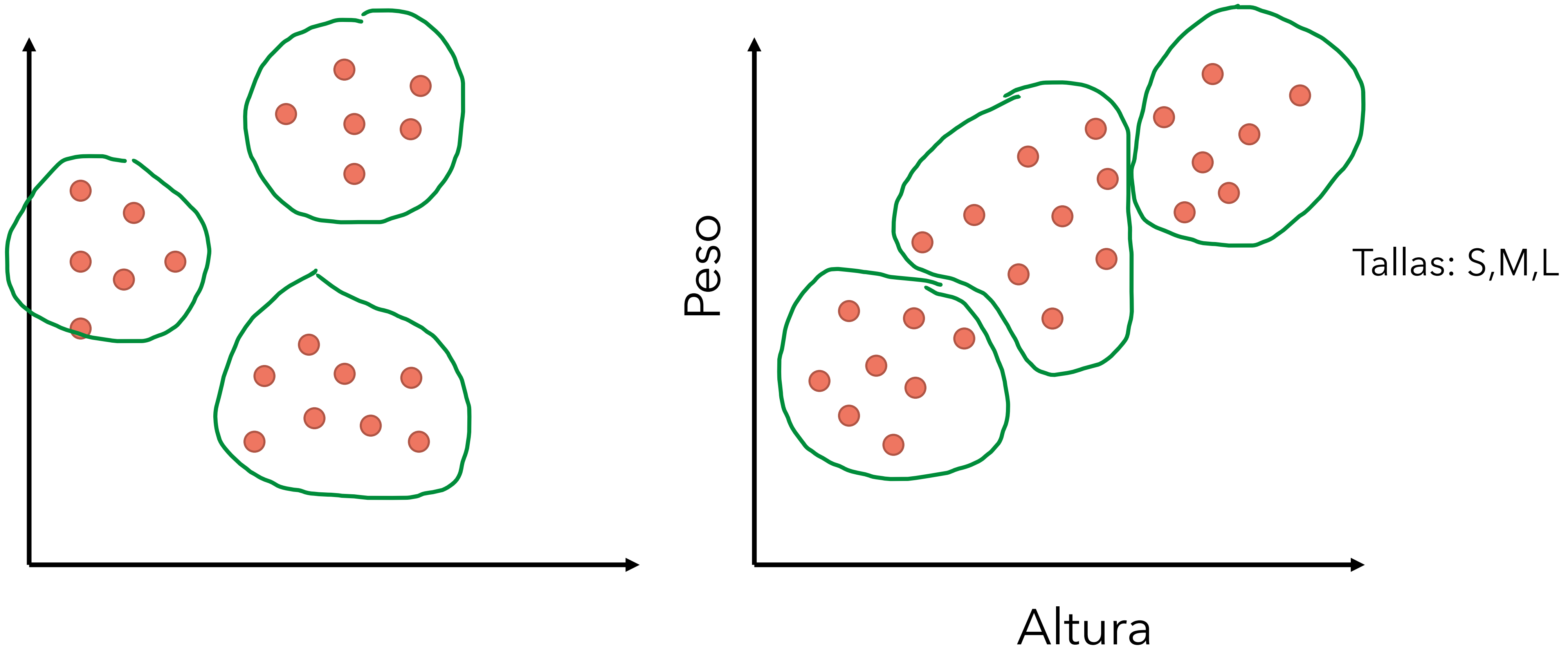
K-Means



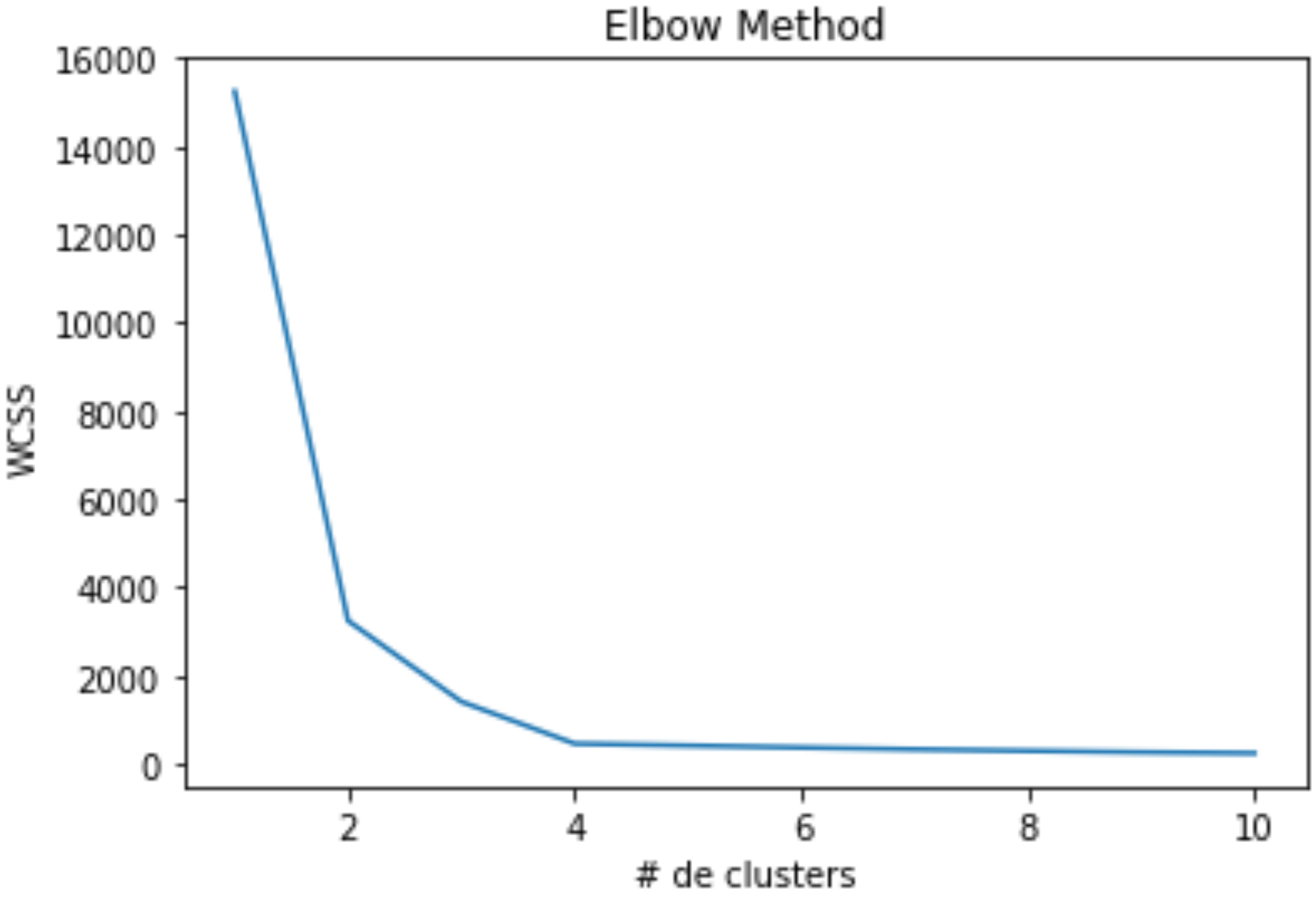


Tallas: S,M,L

K-Means



¿Cómo elegir K?



K-Means ++

Propuesto en 2007 por David Arthur y Sergei Vassilvistkii, es una forma de evitar el problema de malos clústeres que se genera durante el algoritmo de K-means tradicional.

Idea principal: elegir **el primer centroide de entre los puntos x_i del conjunto de datos** y **elegir los otros centroides de entre los restantes** con probabilidad proporcional a la distancia entre el punto elegido y el centroide existente más cercano.

K-Means ++

1. Elegir el primer centroide de entre los puntos x_i aleatoriamente con una distribución uniforme.
2. Para cada punto no elegido, calcular la distancia $D(x)$ entre el punto y el centroide más cercano.
3. Elegir un nuevo centroide, donde se elige x_i con probabilidad proporcional a $D(x)^2$.
4. Repetir 2 y 3 hasta que se elijan k centroides.
5. Proceder con el algoritmo de K-Means tradicional.

K-Means ++

Esto genera un *problema*:

K-Means++ es computacionalmente más caro (tardado), pero el tiempo de convergencia mejora drásticamente, ya que los puntos yacen inicialmente en los posibles clústeres.

K-Means ++

The k-means problem is solved using either Lloyd's or Elkan's algorithm.

The average complexity is given by O(k n T), where n is the number of samples and T is the number of iteration.

The worst case complexity is given by O(n^(k+2/p)) with n = n_samples, p = n_features. (D. Arthur and S. Vassilvitskii, 'How slow is the k-means method?' SoCG2006)

In practice, the k-means algorithm is very fast (one of the fastest clustering algorithms available), but it falls in local minima. That's why it can be useful to restart it several times.

K-Means

Tarea:

Leer el paper *“How slow is the k-means method?”*. Hacer un resumen con los resultados principales.

Nota: Para entender qué es complejidad computacional, revisar esto.

Gracias

Luis Zúñiga

luis.zuniga@correo.uia.mx

https://lzun.github.io