Fundamentos de Ciencias de Datos

Semana 13 - Clustering

Aprendizaje No Supervisado

Hasta ahora hemos visto técnicas de clasificación, donde el **aprendizaje es supervisado**

¿Qué significa esto? Que cada instancia tiene una etiqueta (la respuesta) que el modelo usa para aprender

Pero, ¿Qué pasa cuando no tenemos etiquetas?

Clustering

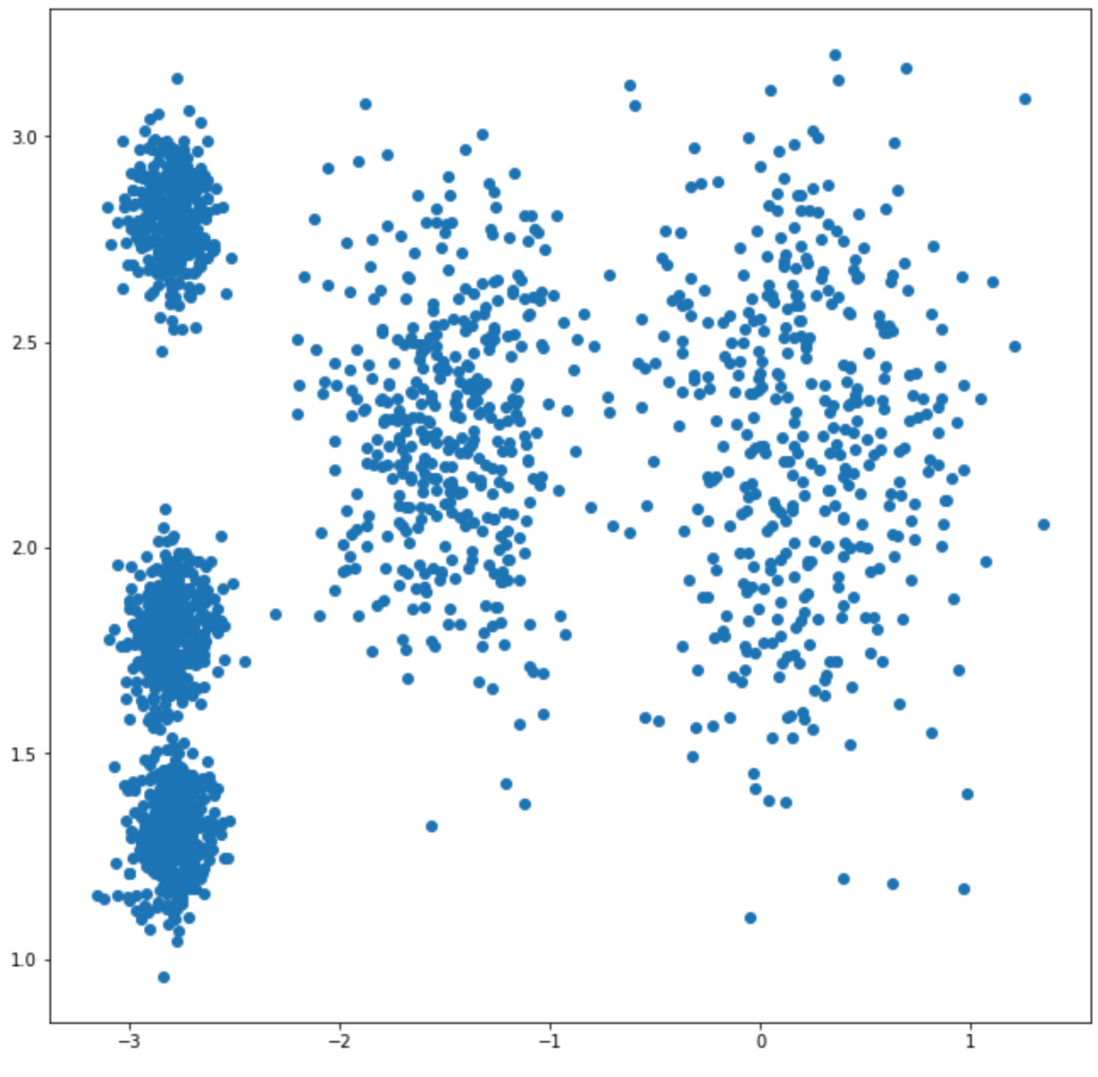
Una de las formas más famosas de hacer aprendizaje no supervisado es hacer **clustering**

Clustering es la tarea de formar grupos a partir de distintos objetos

Dos elementos están en el mismo grupo si son "similares"

Clustering

Ej. ¿Cuántos clusters hay aquí?



Algoritmos de Clustering

Existen varios algoritmos de Clustering, pero en este curso vamos a estudiar dos:

- K-Means
- DBScan

Ejemplo

Dataset de Gatos y perros

Supongamos que tenemos el siguiente *dataset* que guarda el peso y el largo de varias mascotas

Largo (m)	Peso (kg)
0.5	10
1.5	50
1.1	30
...	...

En el *dataset* tenemos datos de gatos y perros, pero no sabemos cuál es cuál

Ejemplo

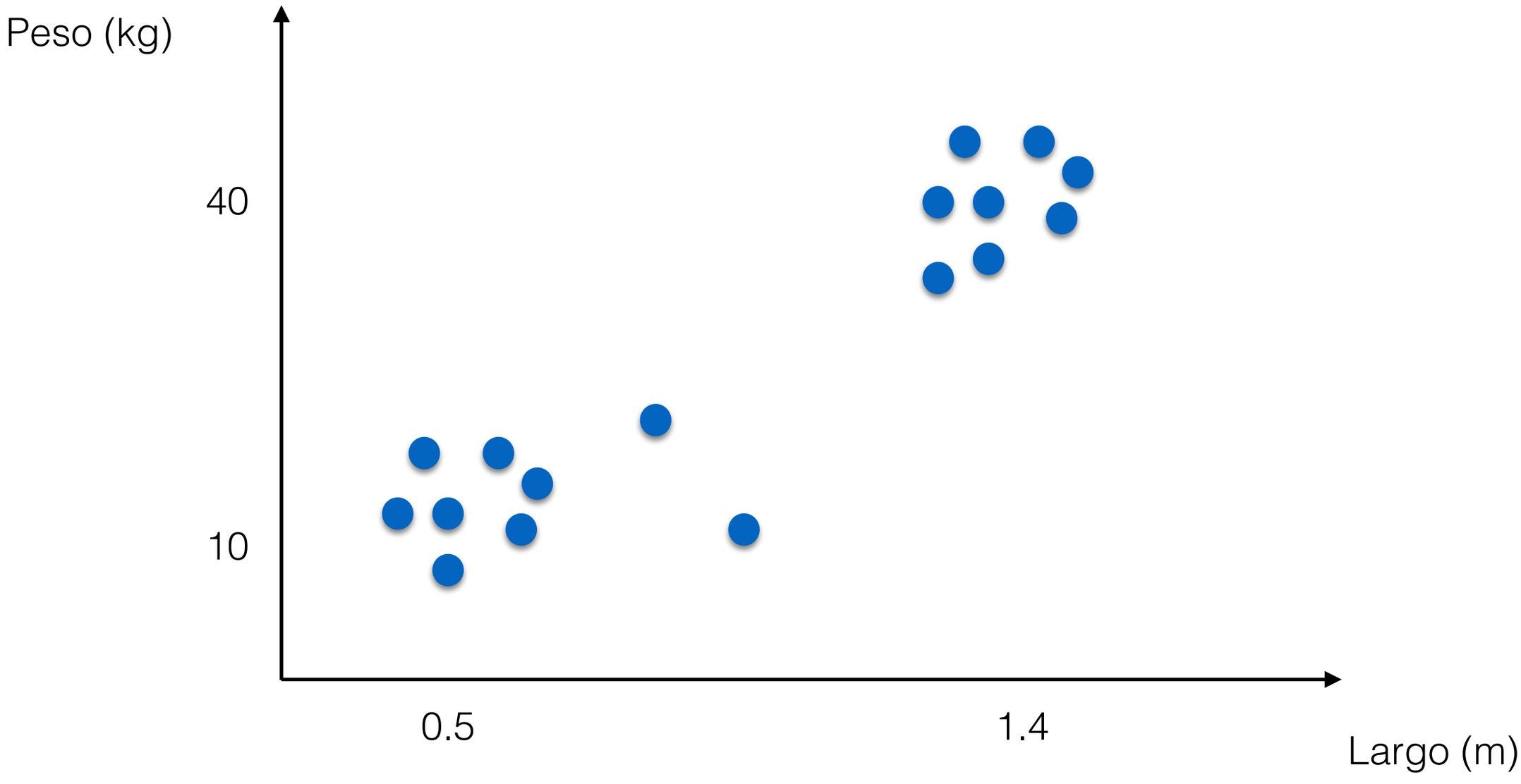
Dataset de Gatos y perros

Vamos a hacer clustering para intentar descubrir los dos posibles grupos

Primero vamos a intentar visualizar los datos

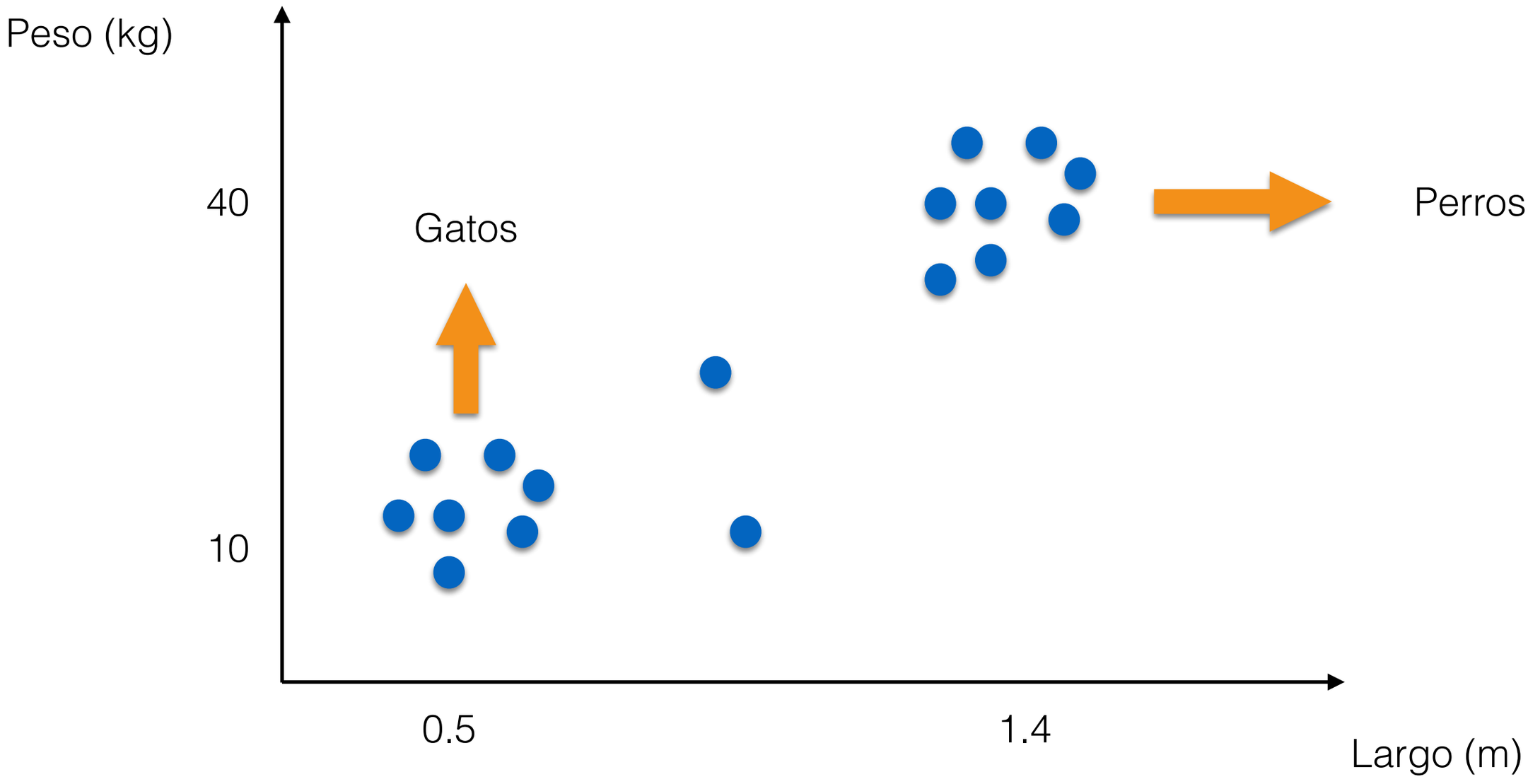
Ejemplo

Dataset de Gatos y perros



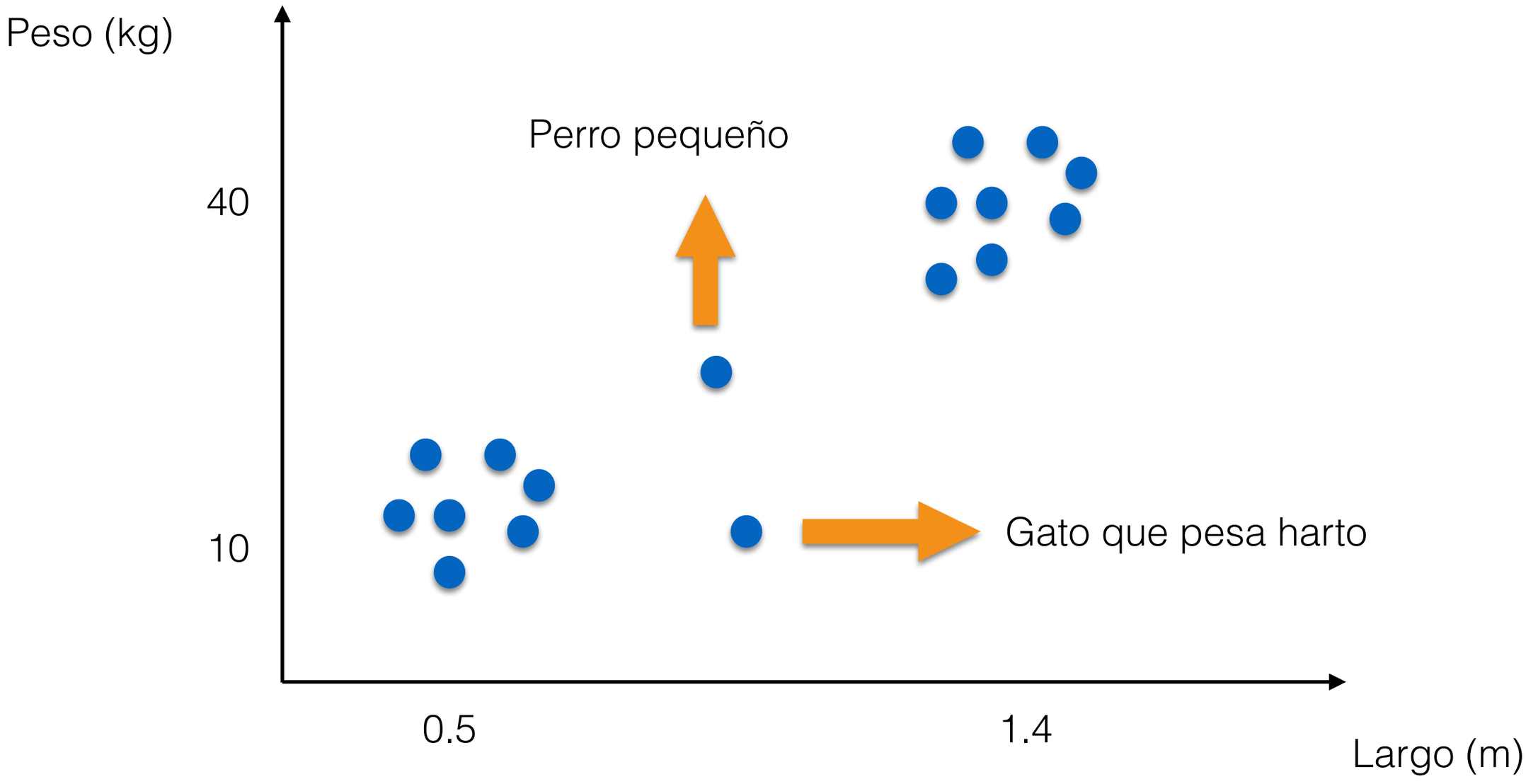
Ejemplo

Dataset de Gatos y perros



Ejemplo

Dataset de Gatos y perros



Ejemplo

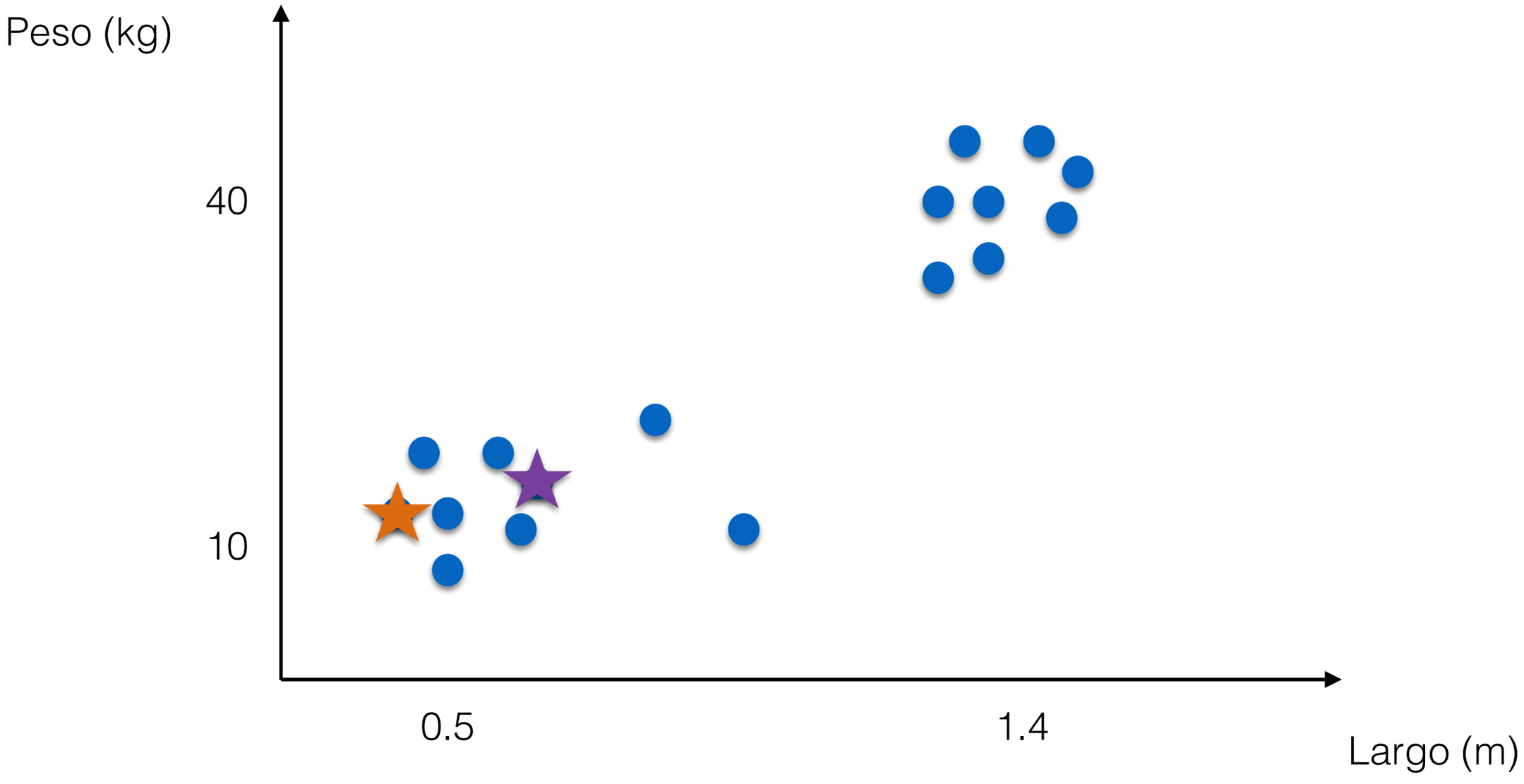
Dataset de Gatos y perros

¿Cómo encontramos nuestros clusters?

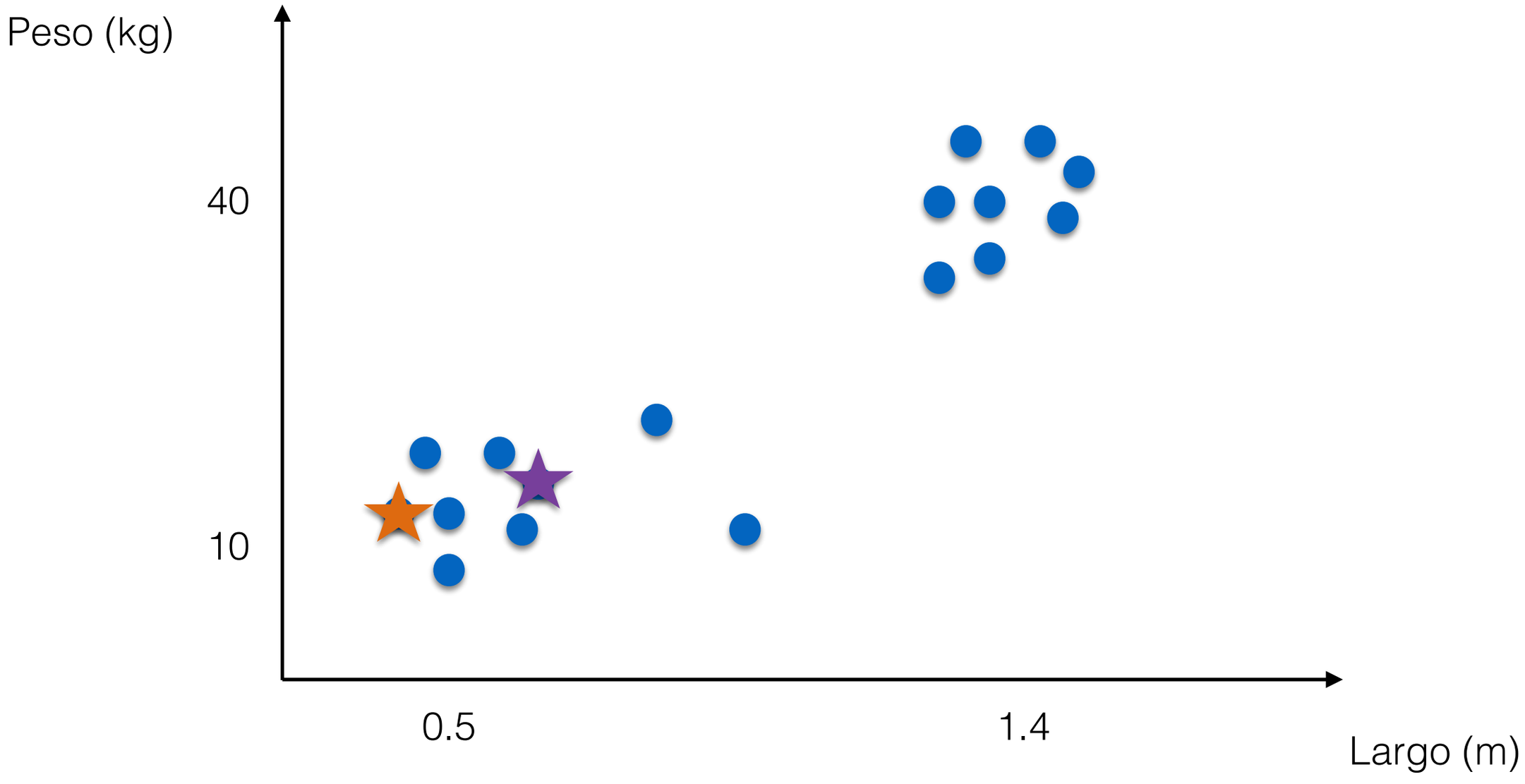
Vamos a explicar como funciona K-Means para encontrar los clusters

Clustering con K-Means

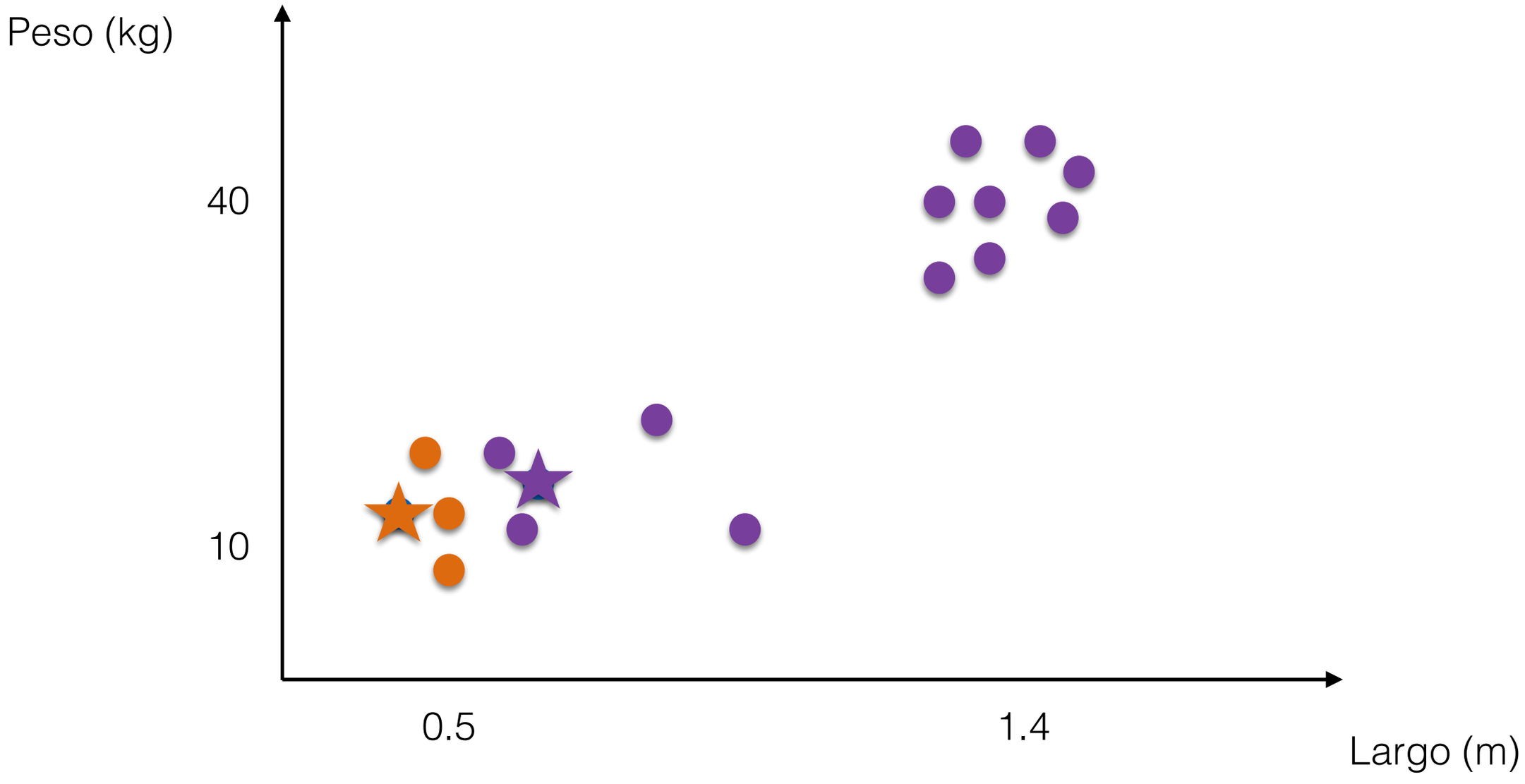
Primero tenemos que inicializar dos "centroides", que corresponde a dos puntos al azar



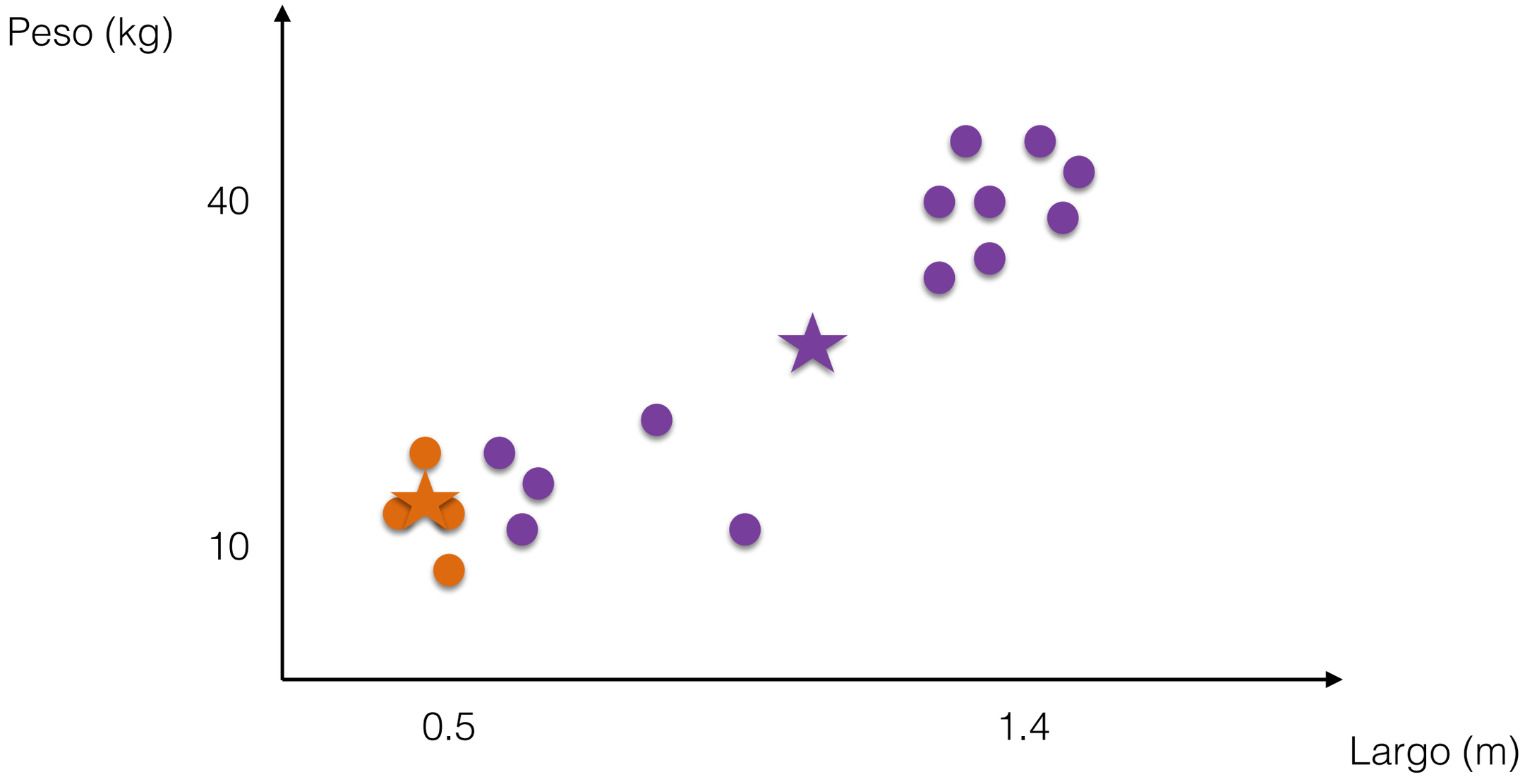
Ahora cada punto va a corresponder al cluster representado por su centroide más cercano



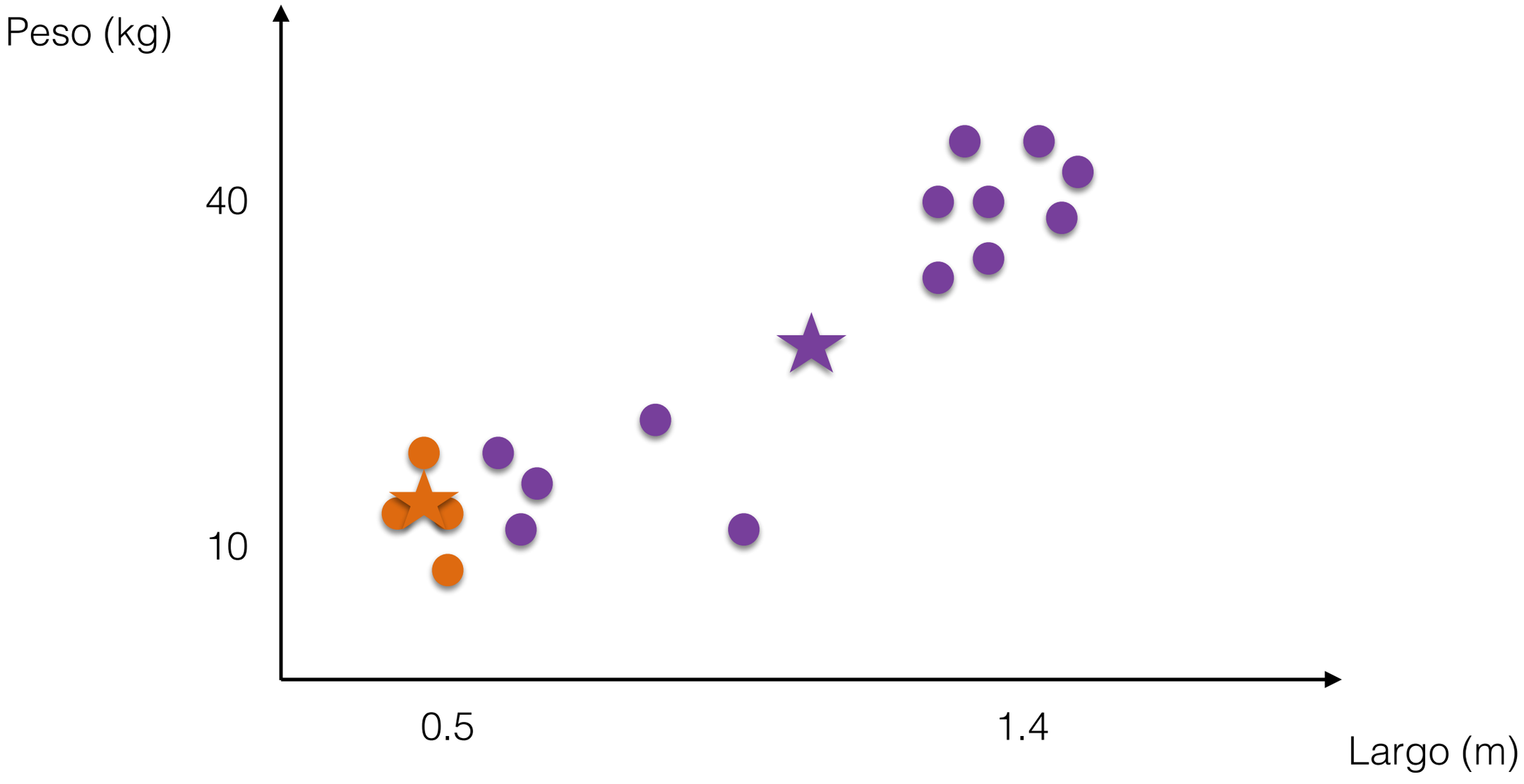
Ahora cada punto va a corresponder al cluster representado por su centroide más cercano



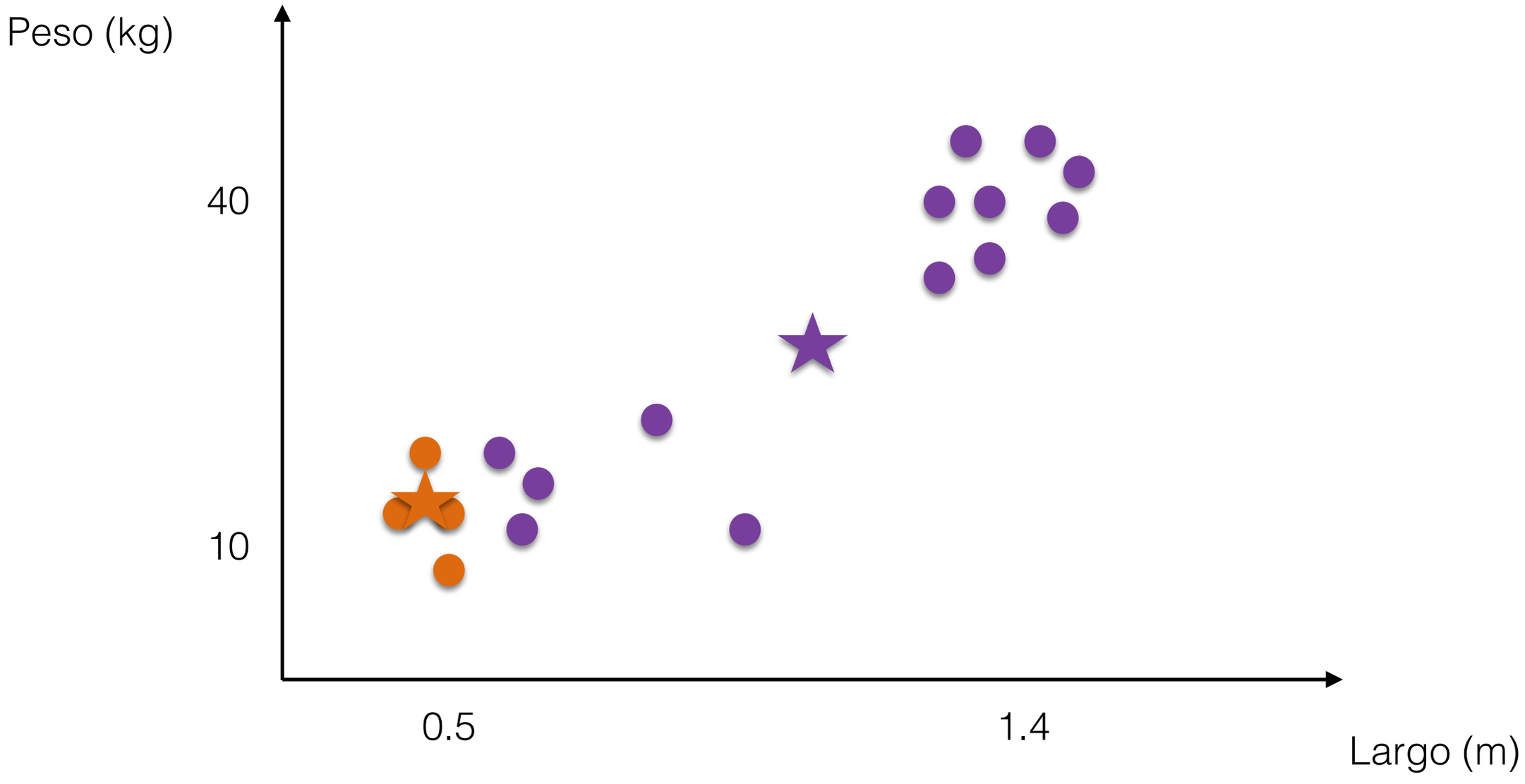
Ahora actualizamos los centroides; **ojo**, la idea de centroide es la misma que deberían recordar de física



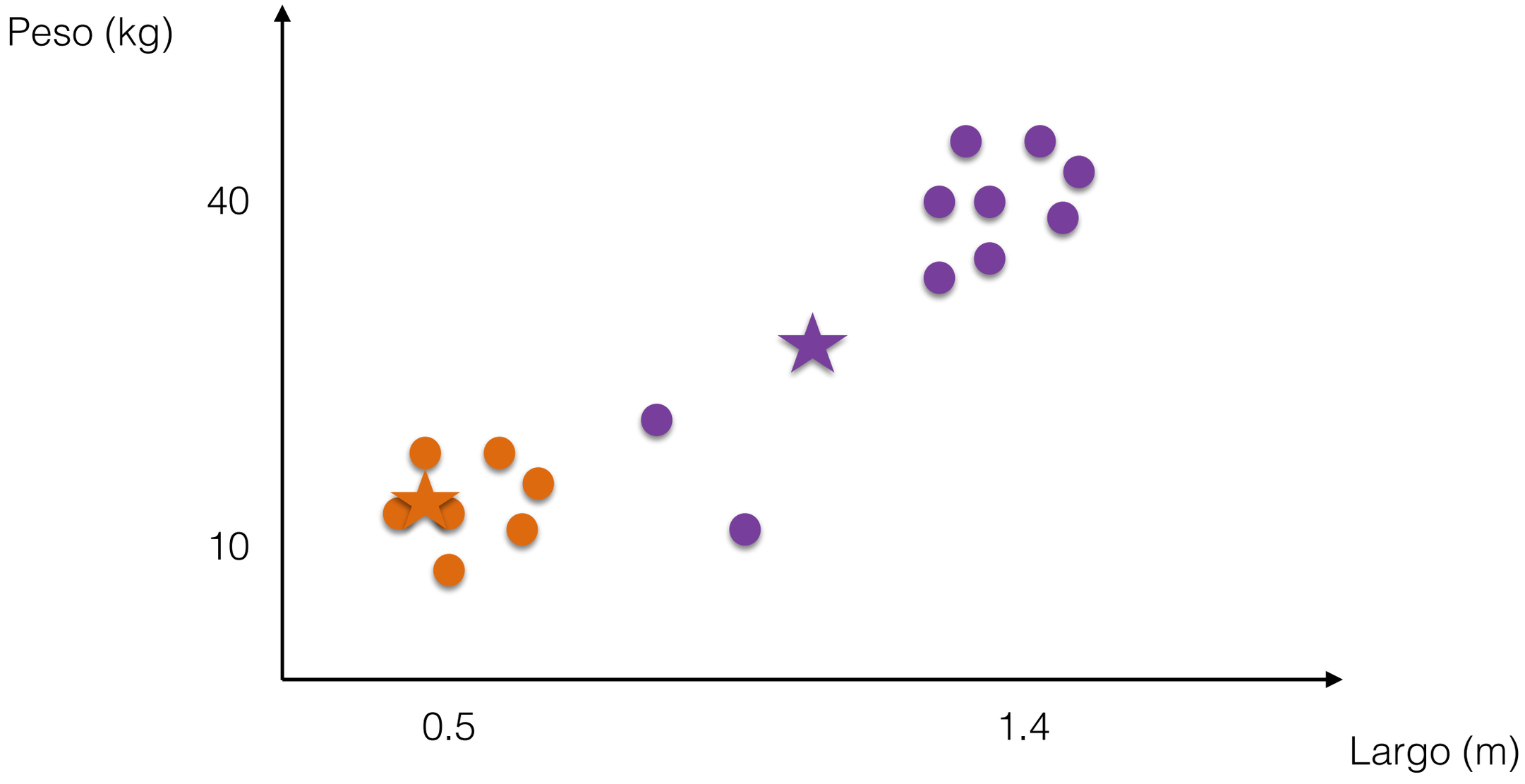
Notemos que los centroides ahora no corresponden a ningún punto en particular



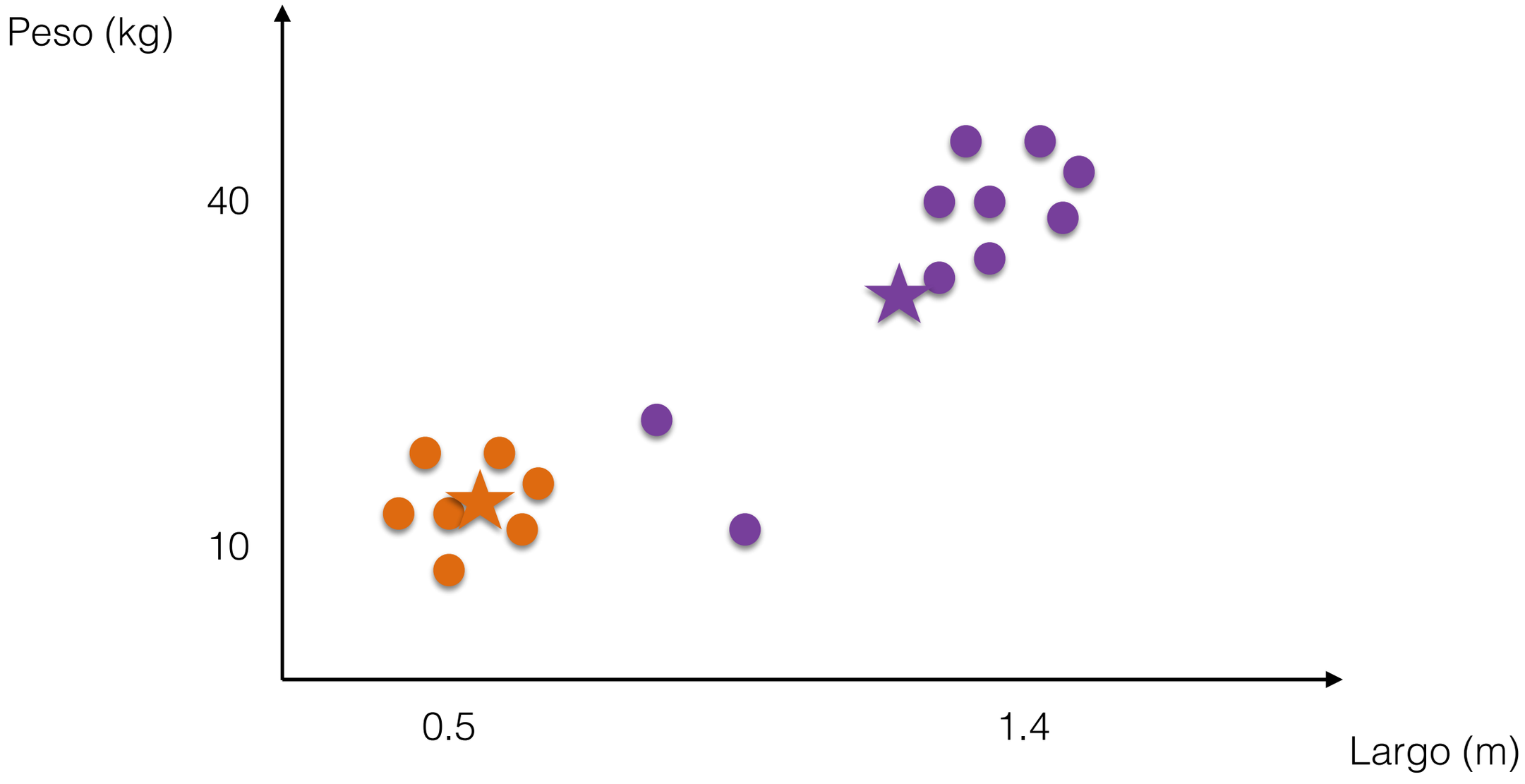
Ahora asignamos cada punto al cluster representado por su centroide más cercano



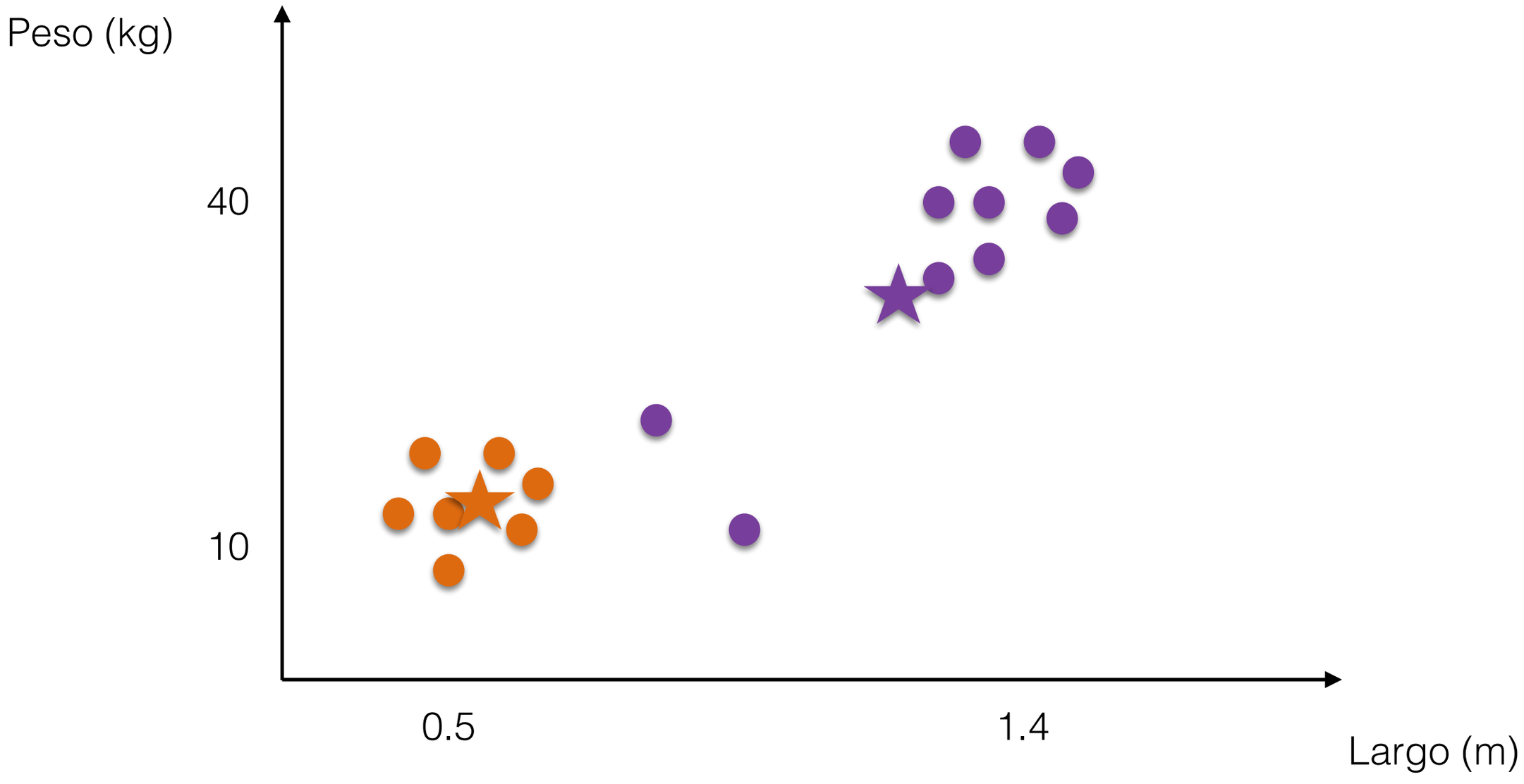
Ahora asignamos cada punto al cluster representado por su centroide más cercano



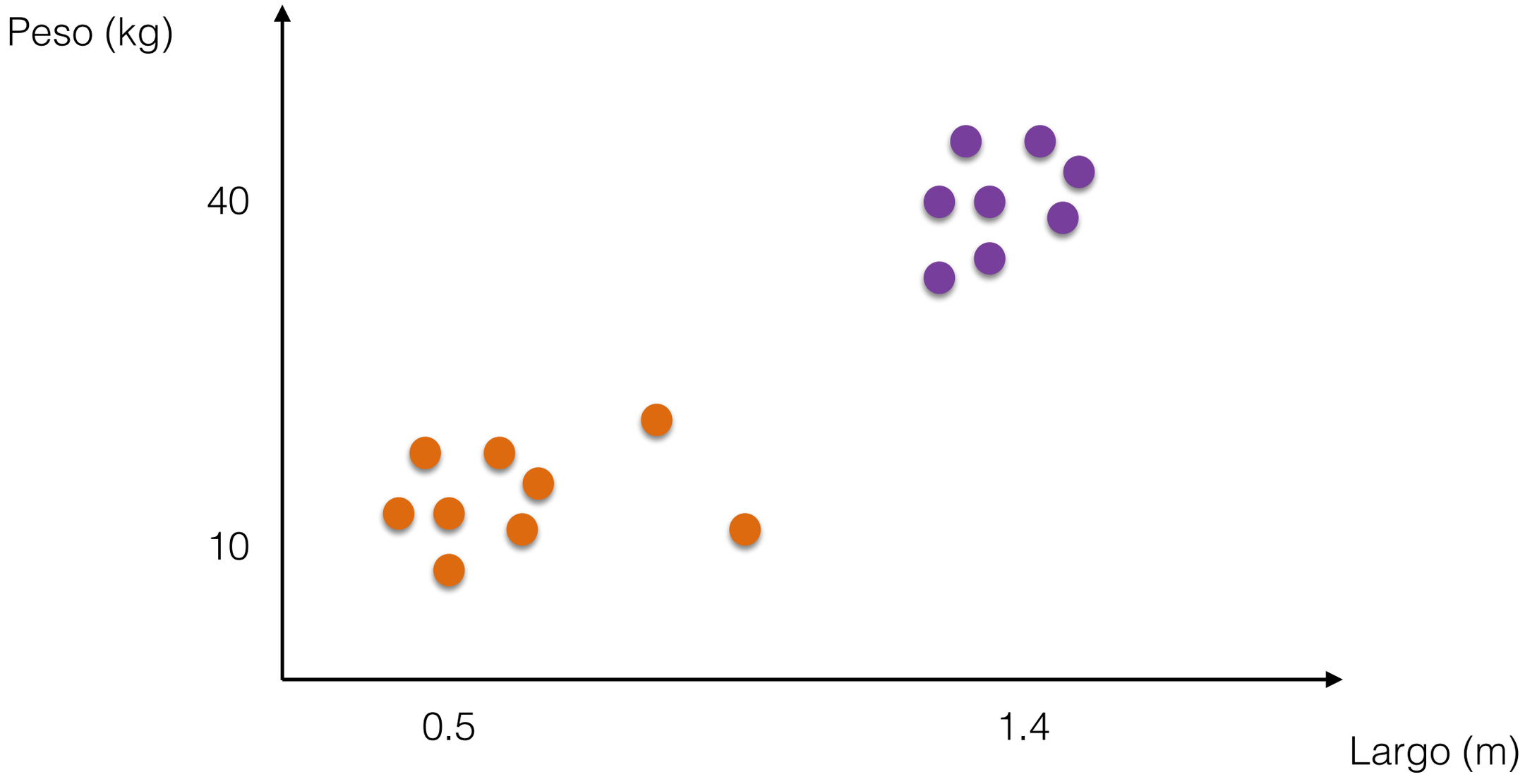
Actualizamos los centroides



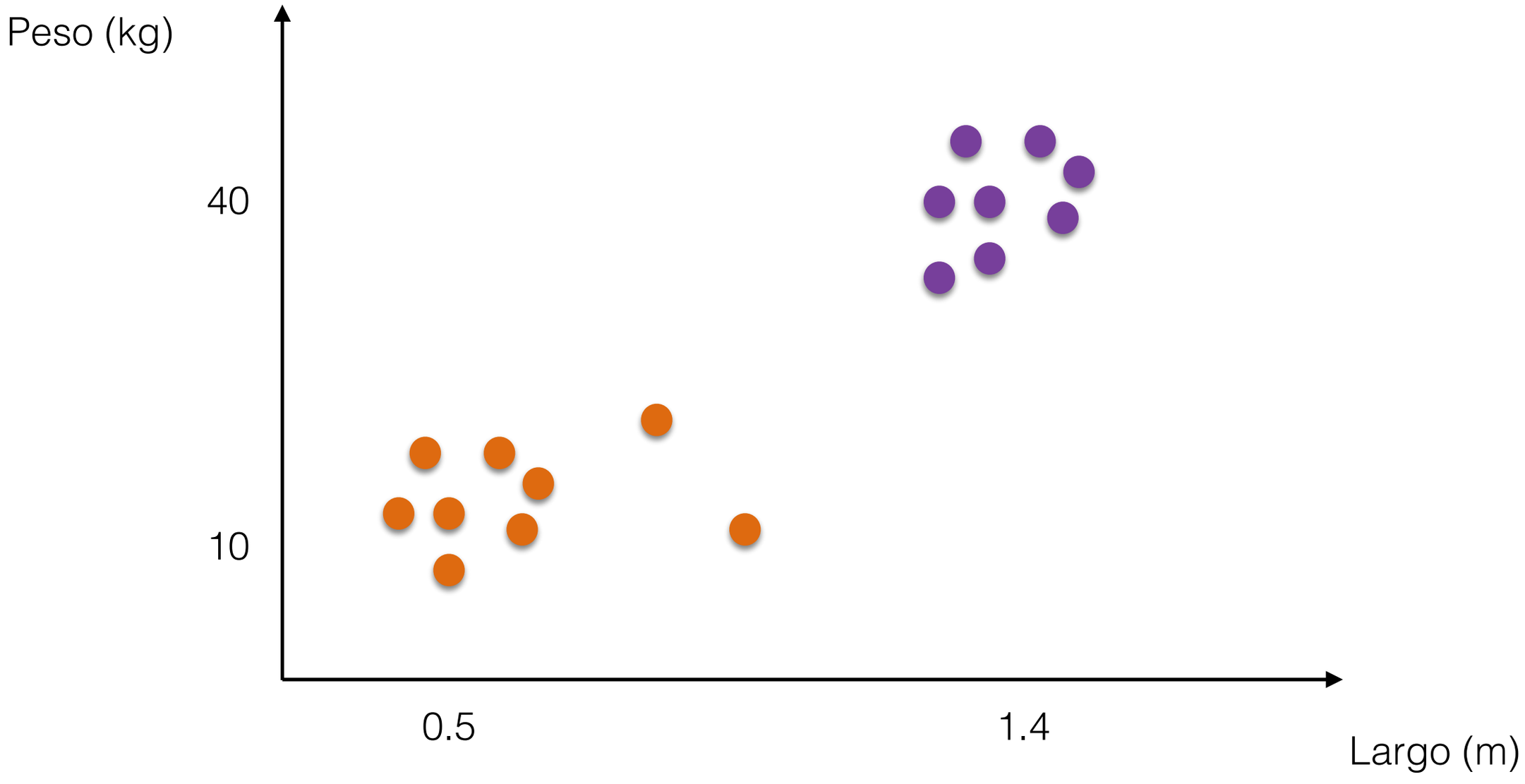
Y seguimos iterando de la misma forma hasta converger



Finalmente encontraremos dos clusters: mascotas pequeñas y mascotas grandes



Ojo, en mascotas pequeñas están los dos outliers: el perro pequeño y el gato que pesa harto



K-Means

Detalles

Si nuestros datos tienen más dimensiones, esta idea la podemos extender (ej. usamos distancia euclidánea entre puntos)

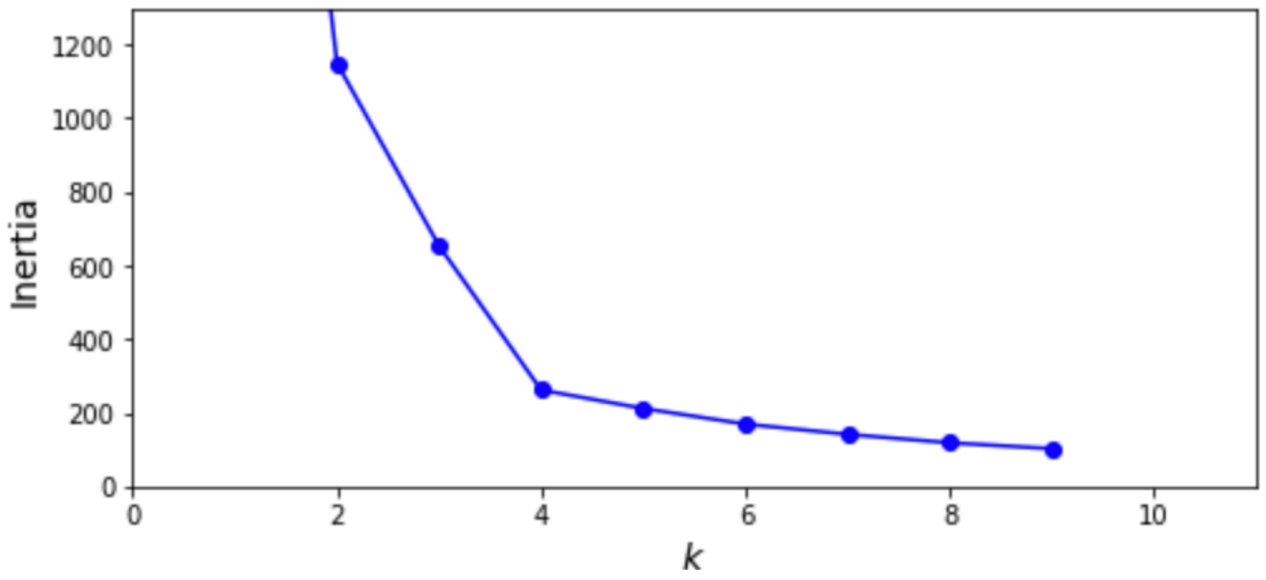
A K-Means le decimos **el número de clusters que queremos encontrar**

¿Qué pasa si no sabemos de antemano el número de clusters?

K-Means

Método del codo

Probamos con distintos números de clusters y utilizamos el método del codo



En el gráfico de arriba el codo está en 4 clusters

DBScan

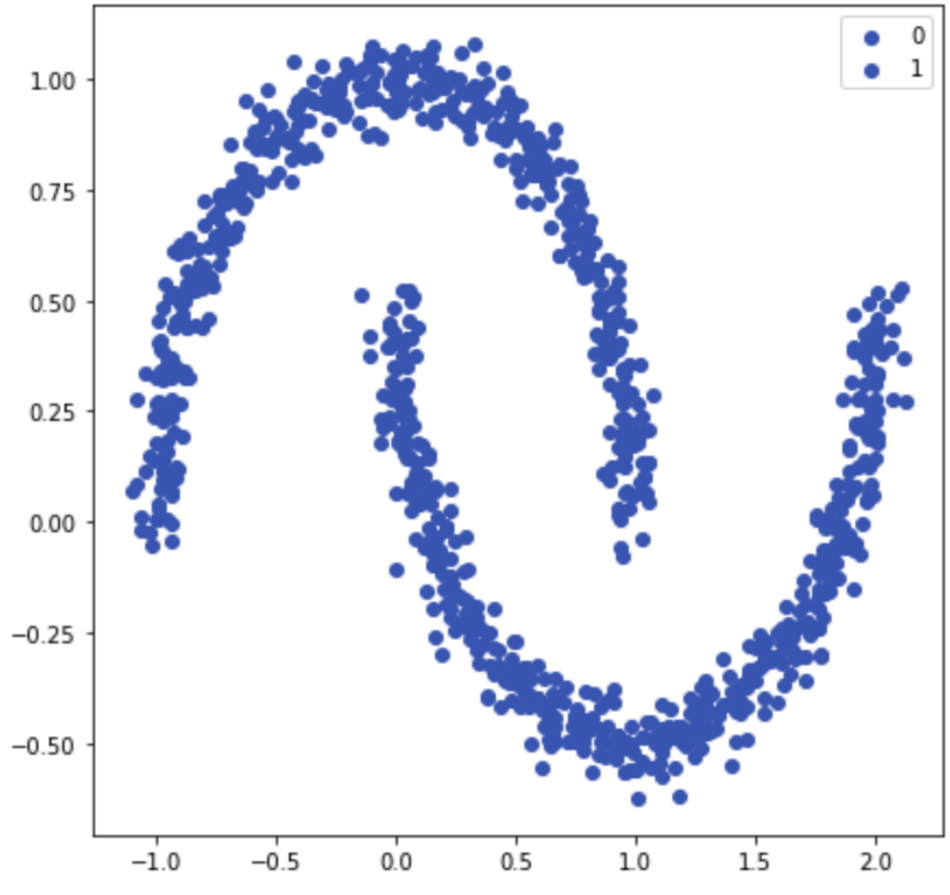
Existe otros algoritmos de reducción de dimensionalidad, como por ejemplo **DBScan**

Este algoritmo funciona de forma distinta: clasifica como un **cluster** zonas que son de alta densidad

DBScan

Ejemplo

¿Cómo crees que funciona el algoritmo K-Means en este caso?



DBScan

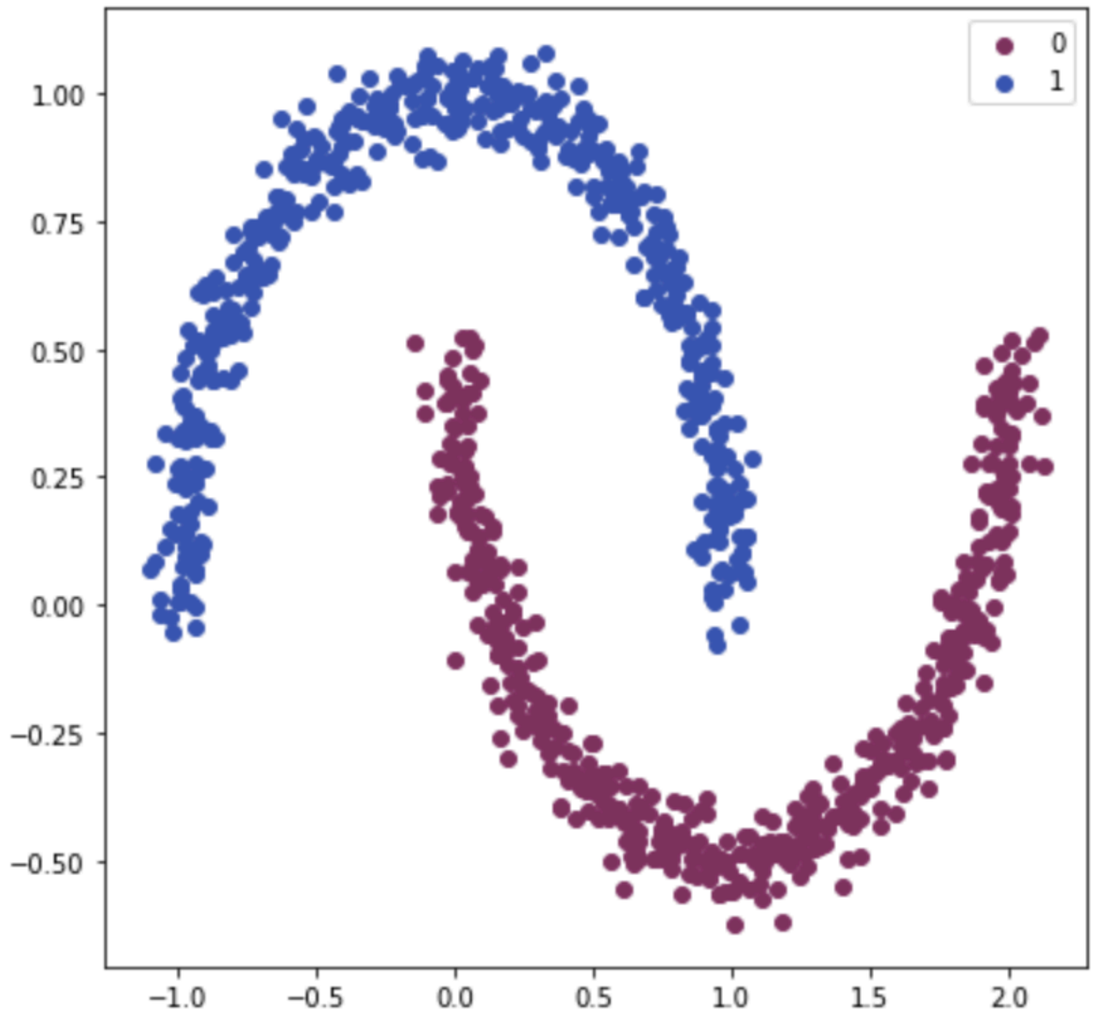
Para manejar estos casos usamos DBScan, algoritmo al que le indicamos la distancia mínima entre dos puntos para ser vecinos

Además le indicamos cuantos vecinos necesitamos para armar un cluster

Así clasificamos como clusters las zonas de alta densidad (i.e. con los elementos con vecinos muy cercanos)

DBScan

Ejemplo



Fundamentos de Ciencias de Datos

Semana 13 - Clustering