

Departamento de Física Médica - Centro atómico Bariloche - IB

Introducción a Machine Learning

Ariel Hernán Curiale
ariel.curiale@cab.cnea.gov.ar

Algunos slides fueron adaptados de Fei Fei Li, J.
Johnson y S. Yeung, cs231n, Stanford 2017.



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



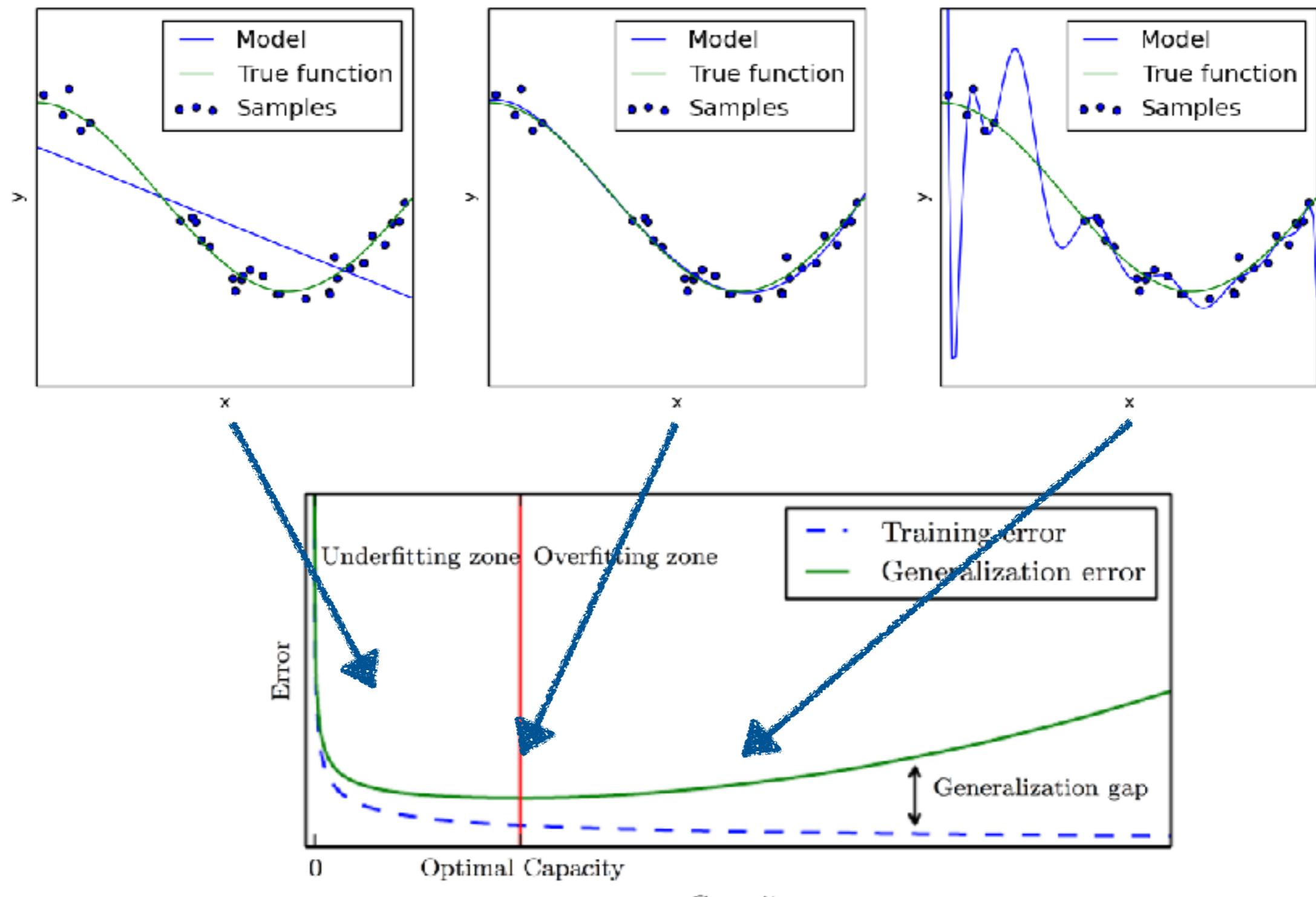
Machine Learning

- ❖ Machine learning: métodos que **aprenden** una tarea a partir de experiencia (datos)
- ❖ Aplicaciones:
 - ❖ Regresión (aproximación de funciones) : se intenta predecir qué valor le corresponde a una entrada
 - ❖ Clasificación: con qué categoría identificamos la entrada
 - ❖ Ej. transcripción de texto o voz, detección de tejido, ...
 - ❖ Clasificación frente a datos incompletos, Reconstrucción de imágenes
 - ❖ Detección de objetos, segmentación, ...
 - ❖ Agrupamiento (clustering)/ categorización: K-means, ...
 - ❖ Super resolución
 - ❖ ...

Conceptos Generales

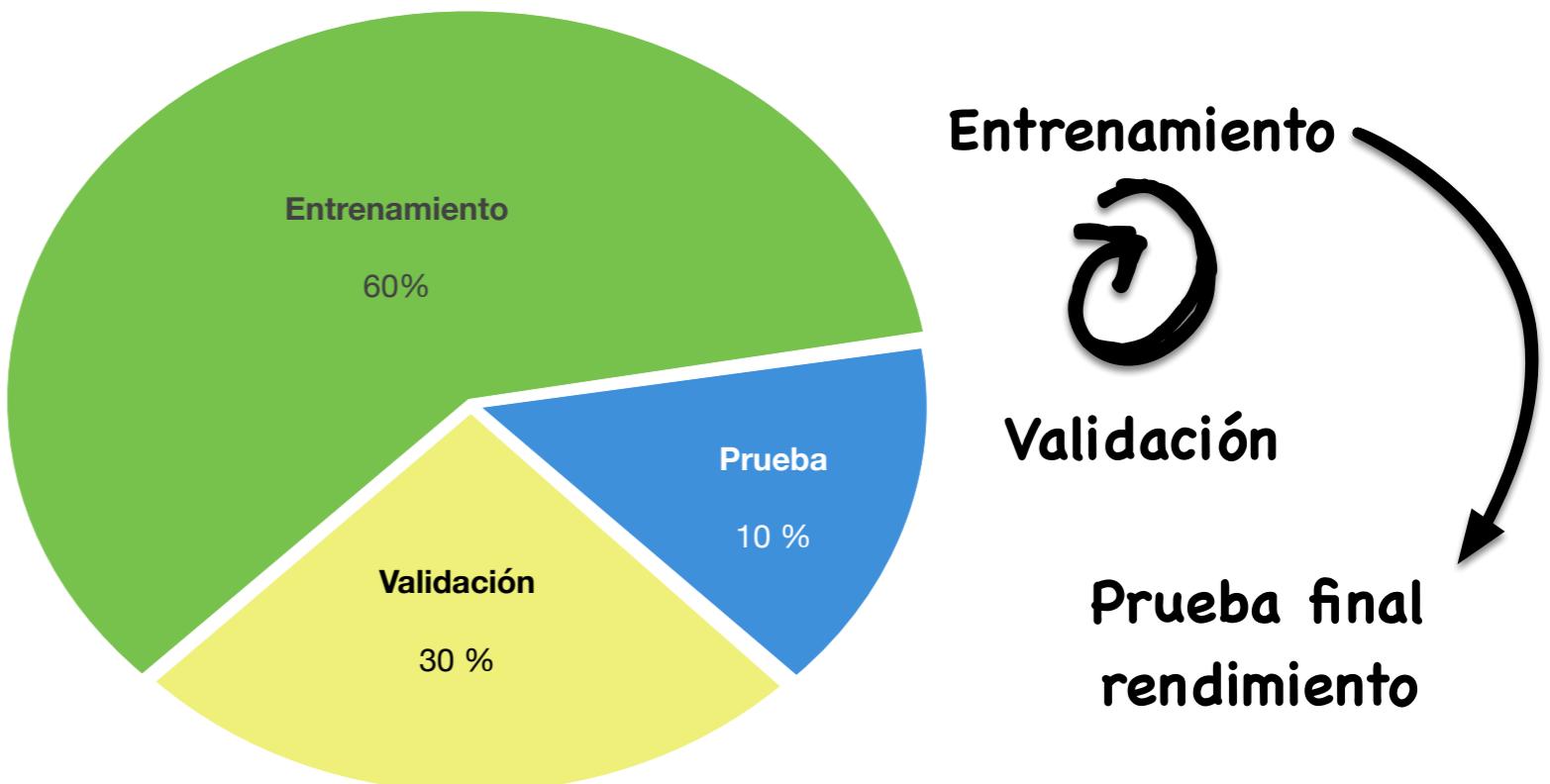


Underfitting - Overfitting



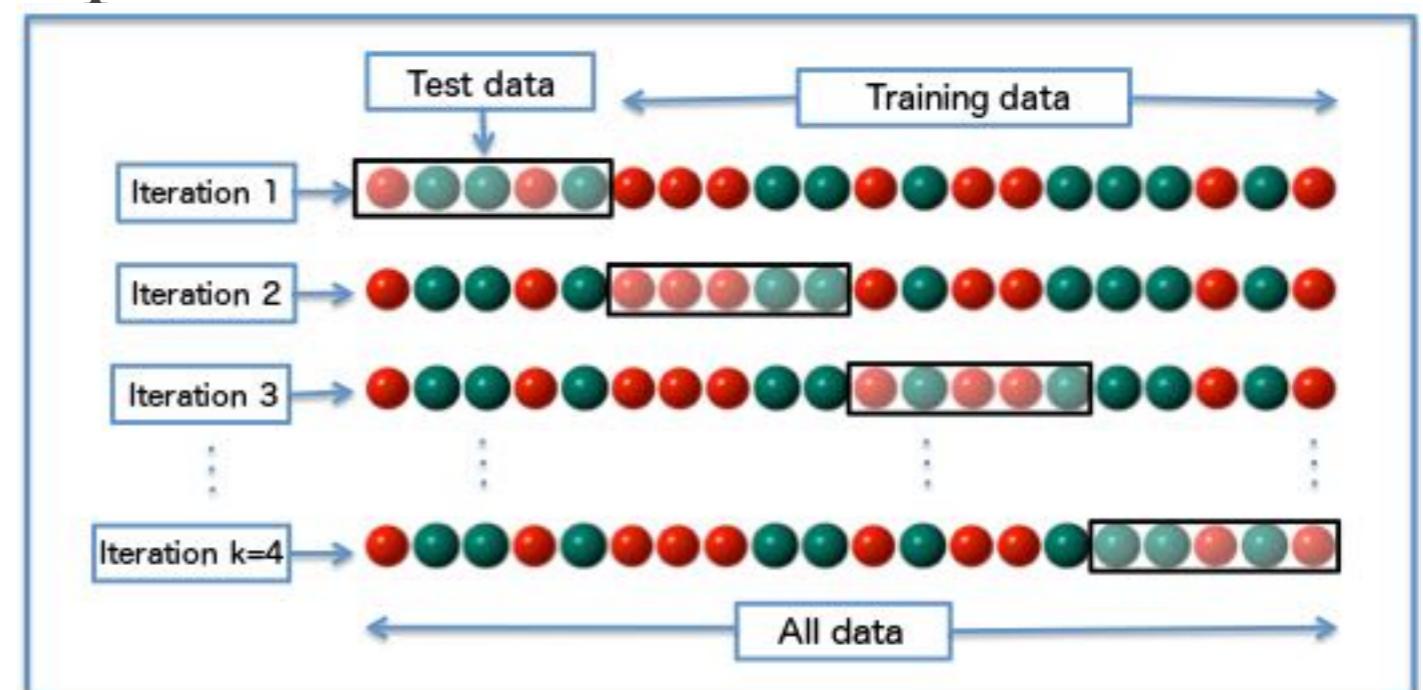
Generalización

- ❖ Nuestro objetivo es Generalizar el conocimiento.
 - ❖ Los datos de prueba no se usan en el entrenamiento, de ninguna forma, ni siquiera para determinar los **hiperparámetros**.
 - ❖ Para evaluar los hiperparámetro usamos un subconjunto de los datos de entrenamiento que llamamos datos de validación.
 - ❖ Validación cruzada.



Validación

- ❖ Existen varias estrategias: k-folding, leave-one-out,
Además hay que ver si es para entrenar hiperparámetros o no.



fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

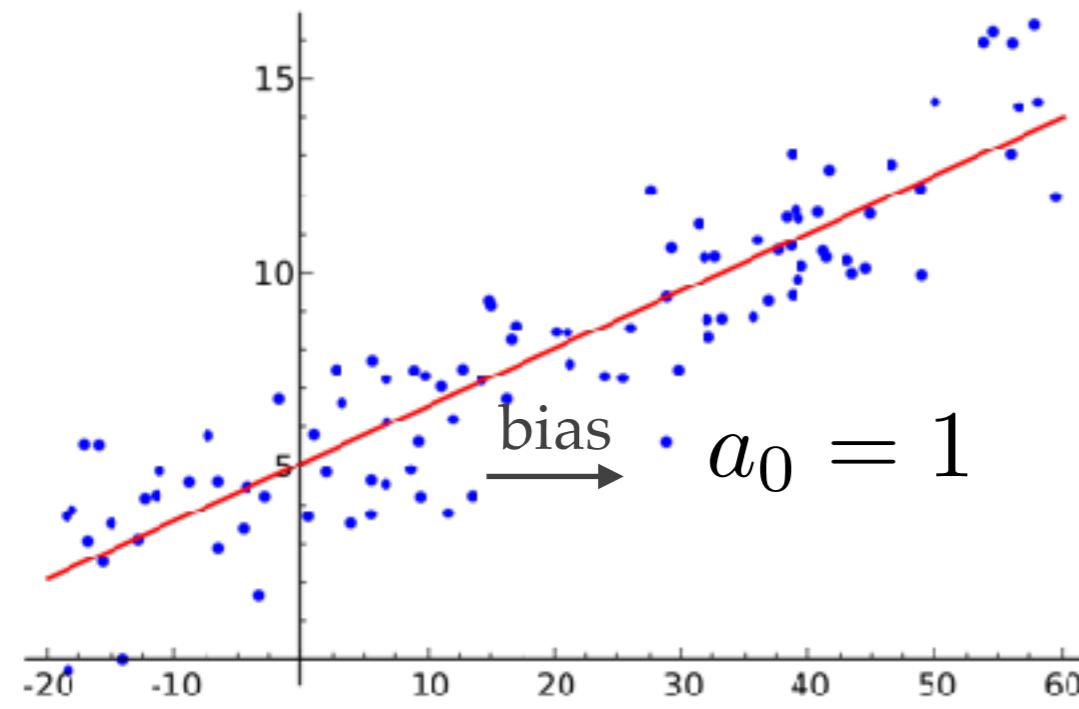
Supervisado y No Supervisado

Veamos dos ejemplos muy simples de técnicas de ML una supervisada y otra no:

- ❖ Una regresión lineal (cuadrados mínimos).
- ❖ K-means clustering

Regresión Lineal (supervisado)

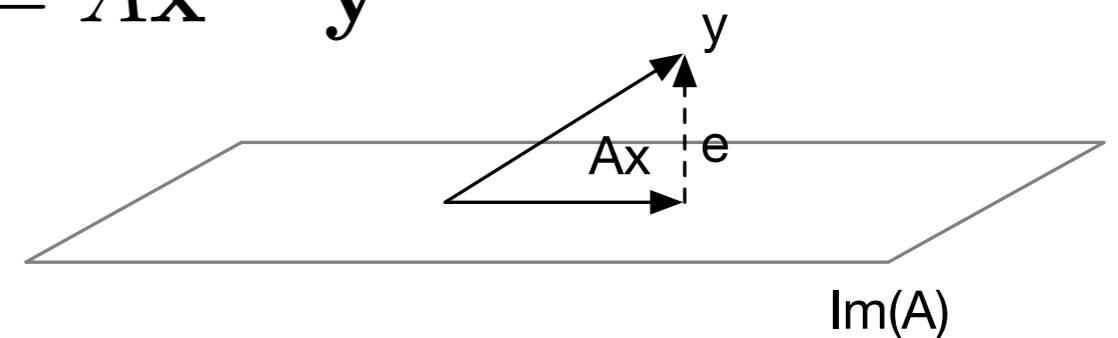
Objetivo: predecir el valor $y \in \mathbb{R}$ dado un conjunto de características $\mathbf{a} \in \mathbb{R}^n$ mediante una relación lineal:



$$y = a_1 x_1 + \cdots + a_n x_n$$

$$\mathbf{y} = A\mathbf{x}$$

$$\mathbf{e} = A\mathbf{x} - \mathbf{y}$$

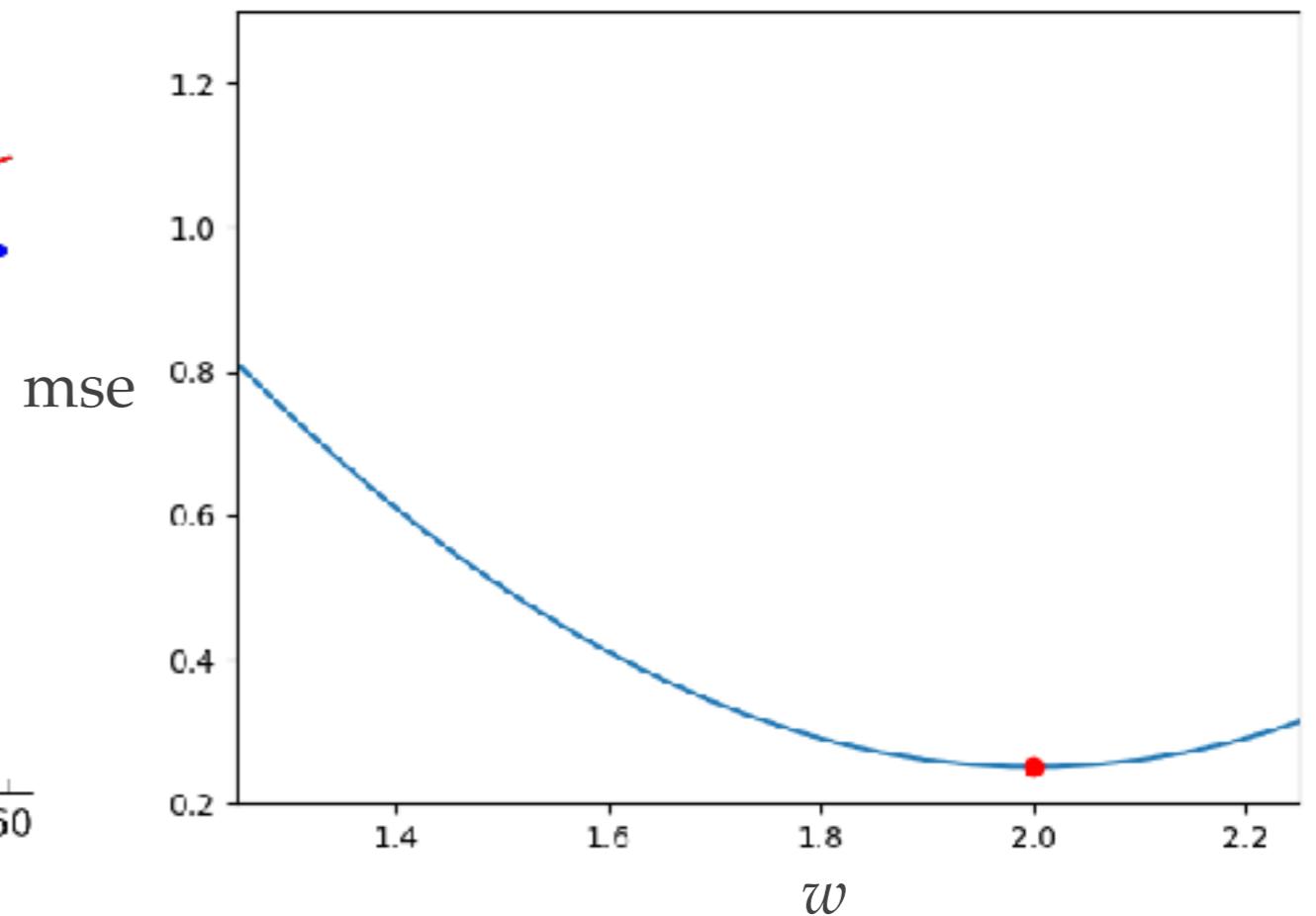
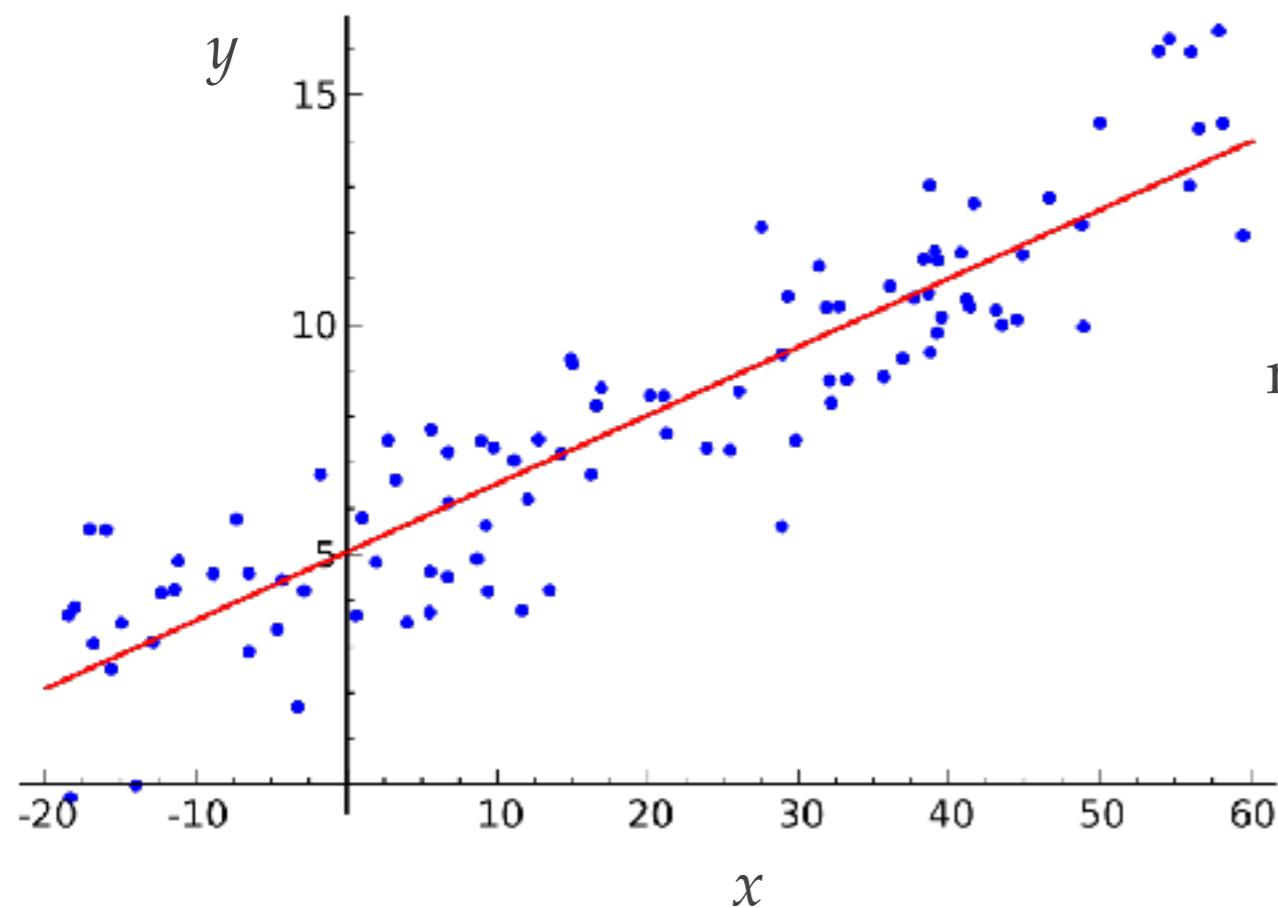


$$\min \|\mathbf{e}\| \Rightarrow \mathbf{e} \perp A\mathbf{s} \Rightarrow \langle \mathbf{e}, A\mathbf{s} \rangle = \mathbf{s}^T A^T (A\mathbf{x} - \mathbf{y}) = 0$$

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{y}$$

Regresión Lineal

¿Qué tiene de bueno esta función de costo?

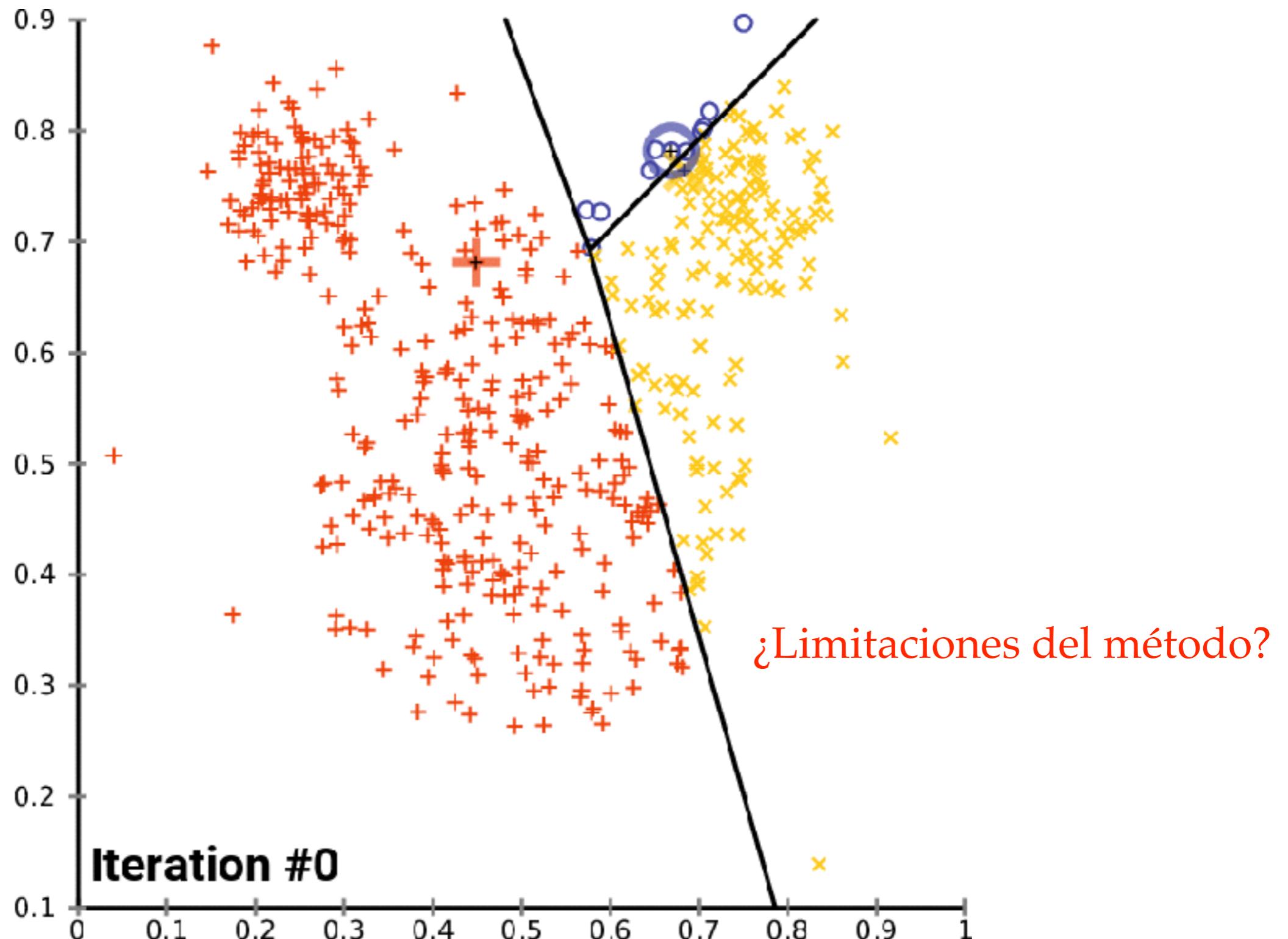


¿Será algo habitual? ¿Qué pasa en espacios de altas dimensiones?

K-means (no supervisado)

- ❖ Agrupamos n muestras en k clases o agrupaciones (clusters).
 - ❖ Para ello minimizamos la suma de distancias intra-clase al cuadrado:
$$\min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \mu_i\|^2$$
- ❖ Método:
 - ❖ ¿Cuales son los hiperparámetros?
 - ❖ Partimos de un conjunto de medias μ_1, \dots, μ_k
 - ❖ Asignamos cada muestra a una agrupación (cluster) que tiene la media más cercana
 - ❖ Se vuelve a calcular la media para cada agrupación (cluster)
 - ❖ Iteramos hasta un estado de equilibrio

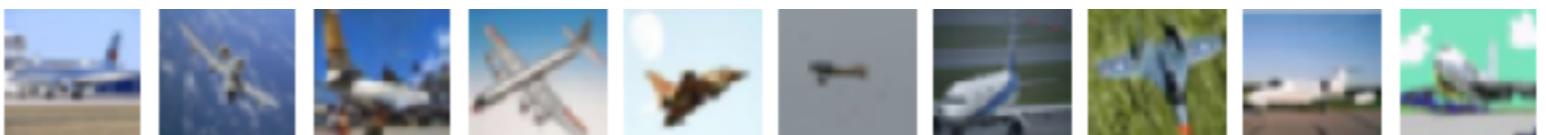
K-means



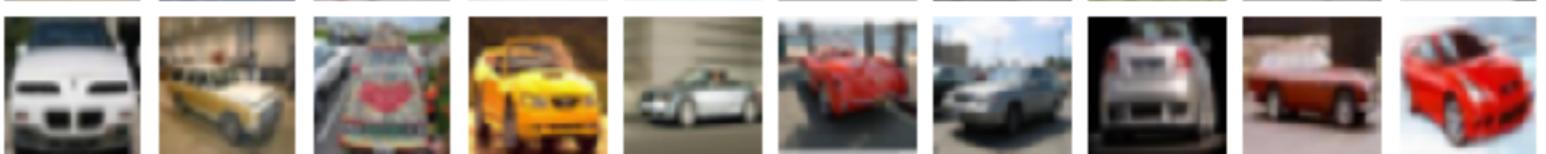
Clasificación de imágenes

- ❖ CIFAR-10:
 - 10 clases
 - 50k training
 - 10k test

airplane



automobile



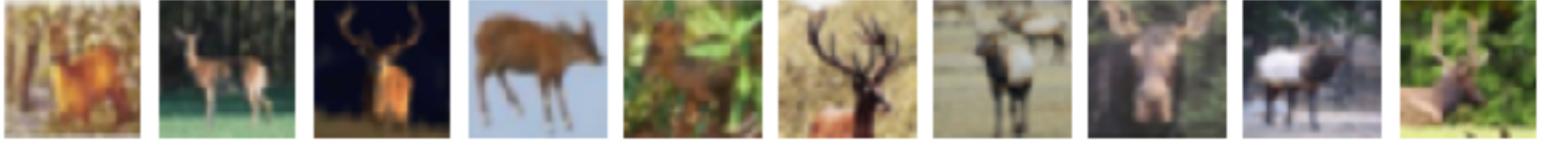
bird



cat



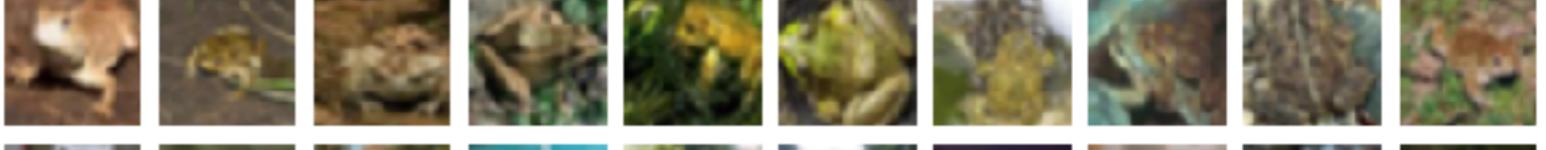
deer



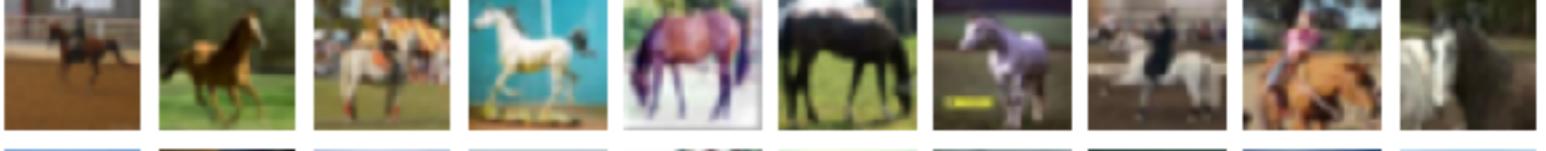
dog



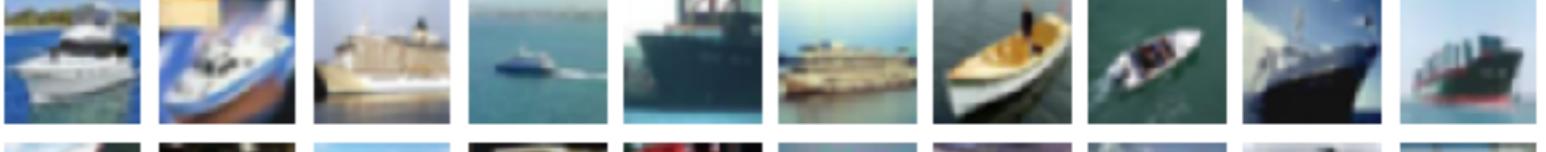
frog



horse



ship



truck



Desafíos



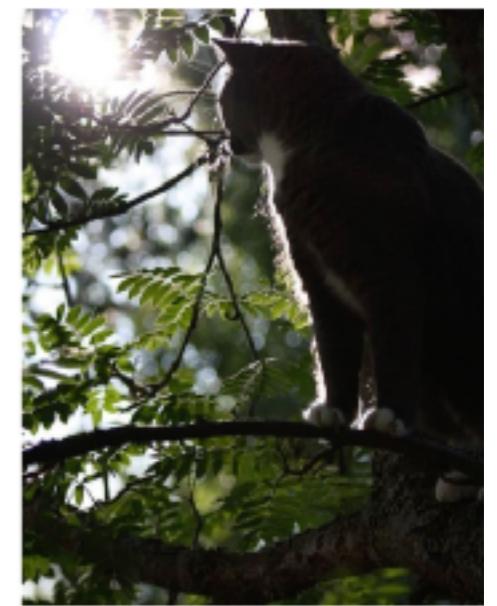
This image is CC0 1.0 public domain



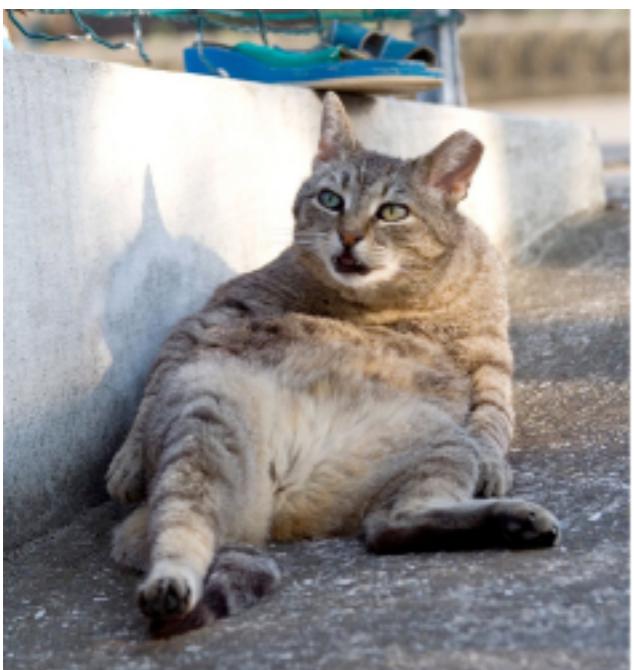
This image is CC0 1.0 public domain



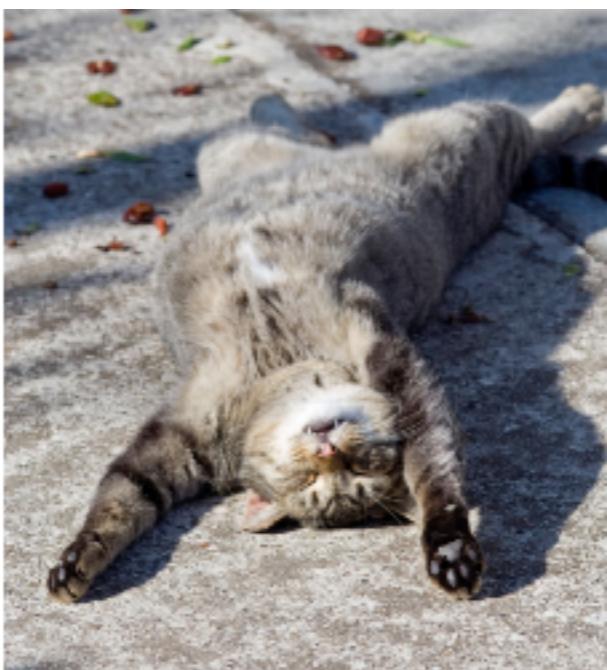
This image is CC0 1.0 public domain



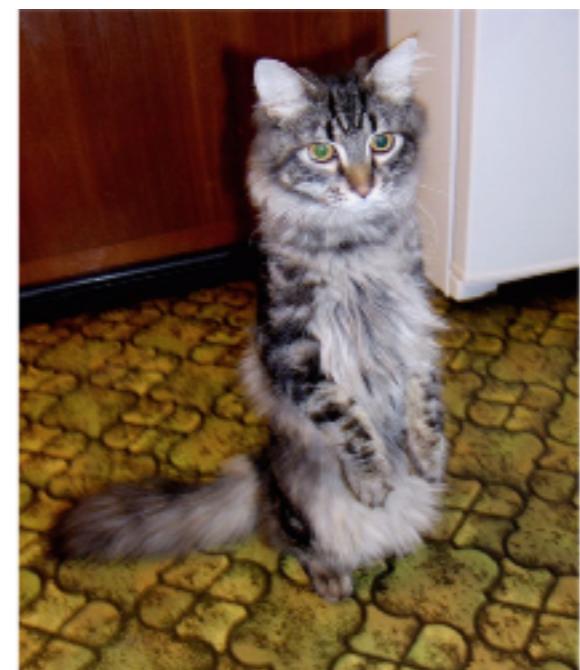
This image is CC0 1.0 public domain



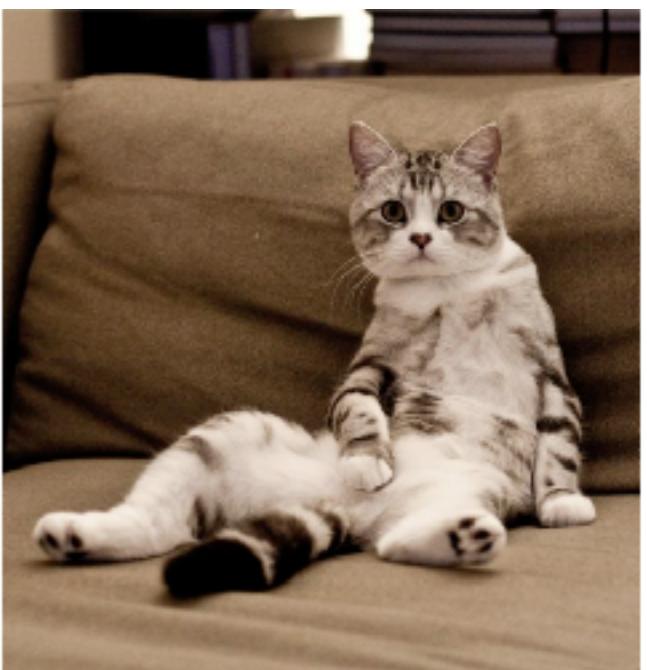
This image by Umberto Salvagnin
is licensed under CC-BY 2.0



This image by Umberto Salvagnin
is licensed under CC-BY 2.0



This image by sare bear
is licensed under CC-BY 2.0



This image by Tom Thaiss
is licensed under CC-BY 2.0

Desafíos



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image by jonsson is licensed under CC-BY 2.0](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)

Clasificación de imágenes

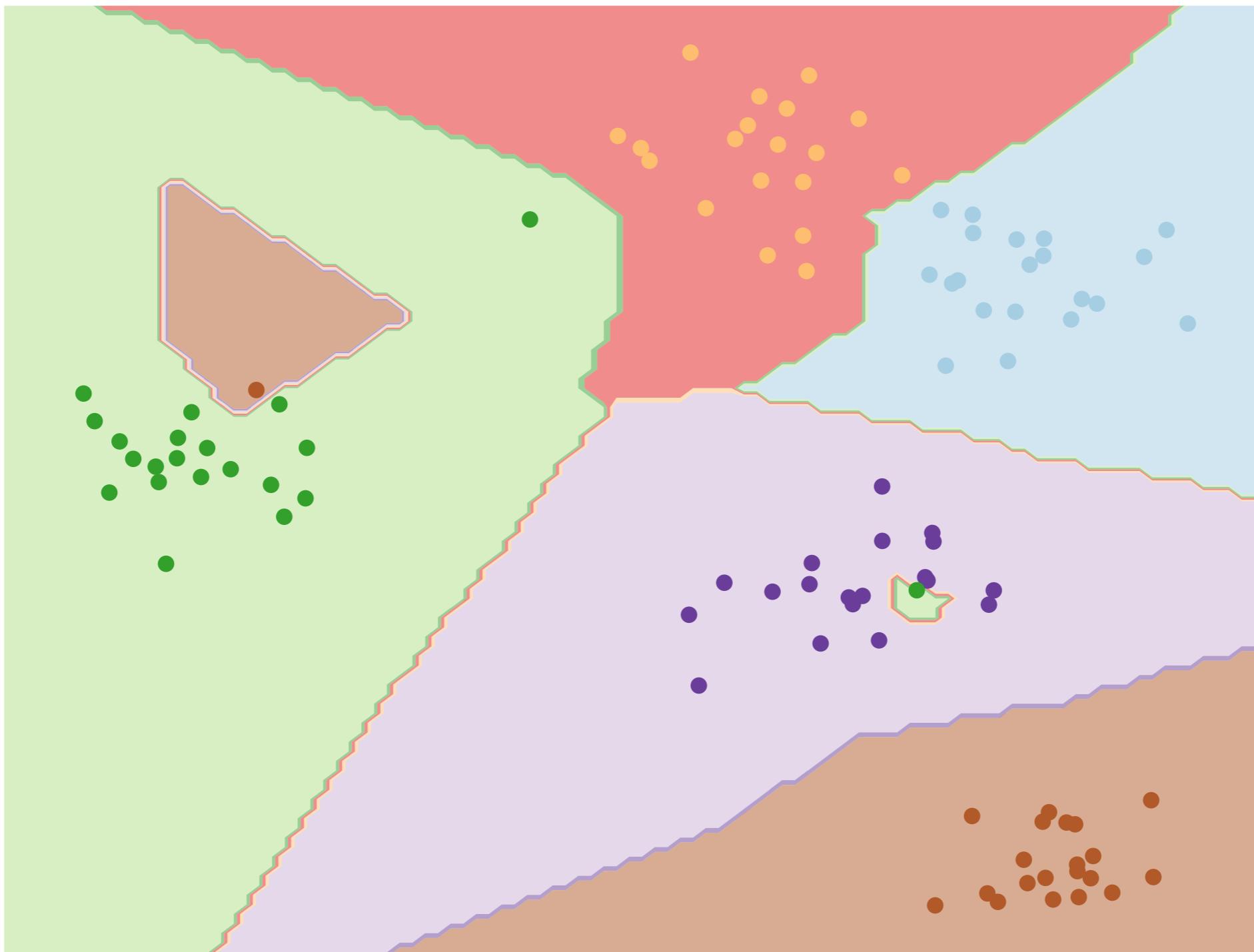
- ❖ Nearest Neighbors
- ❖ K-Nearest Neighbors
- ❖ Linear Classifier :
 - ❖ Support Vector Machine
 - ❖ Softmax Classifier

Clasificación de imágenes: Nearest Neighbors

- ❖ Nearest Neighbors : Vamos a clasificar nuestro ejemplo con la misma clasificación del ejemplo de entrenamiento cuya similitud es la máxima

```
46 from keras.datasets import cifar10
45 import numpy as np
44
43
42
41 class NearesNeighbor:
40
39     def __init__(self):
38         self.X = None
37         self.Y = None
36
35     def train(self, X, Y):          Almacenamos los datos
34         # Las img vienen en (n, row,col,rgb)
33         self.im_shape = X.shape[1:]
32         self.X = np.reshape(X, (X.shape[0], np.prod(self.im_shape)))
31         self.Y = Y
30
29     def predict(self, X):
28         assert self.X is not None, 'Train method needs to be call first'
27         Yp = np.zeros(X.shape[0], np.uint8)
26         for idx in range(X.shape[0]):
25             norm = np.linalg.norm(self.X - X[idx].ravel(), axis=-1)
24             idmin = np.argmin(norm)
23             Yp[idx] = self.Y[idmin]
22
21         return Yp                      Buscamos el más parecido
20
19
18 # num_classes = 10            Cargamos los datos con keras
17
16 # The data, split between train and test sets:
15 (x_train, y_train), (x_test, y_test) = cifar10.load_data()
14 print('x_train shape:', x_train.shape)
13 print(x_train.shape[0], 'train samples')
12 print(x_test.shape[0], 'test samples')
11
10
9 model = NearesNeighbor()
8 model.train(x_train, y_train)
7 yp = model.predict(x_test[:10])
6
```

Frontera de decisión

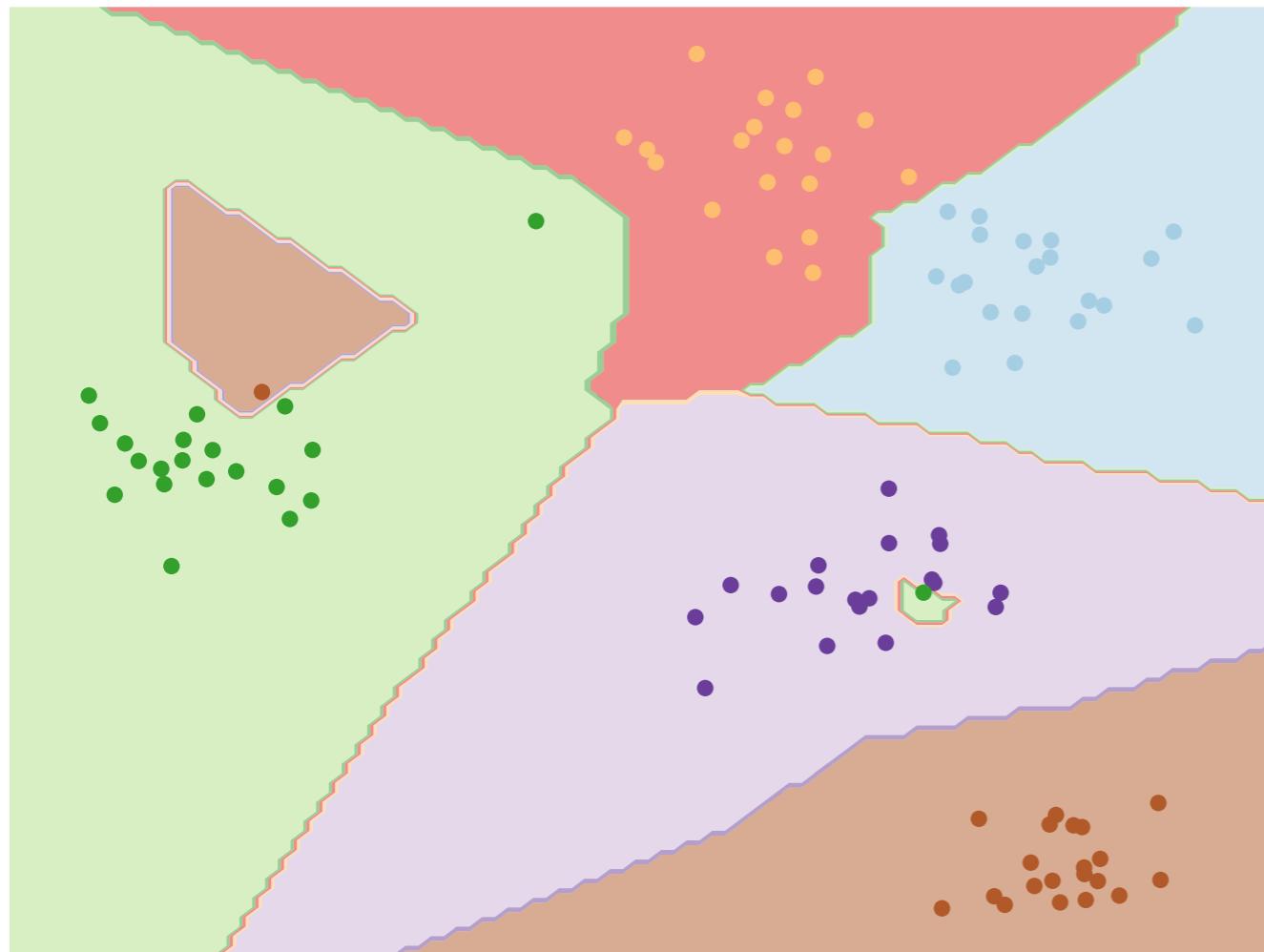


Clasificación de imágenes: KNN

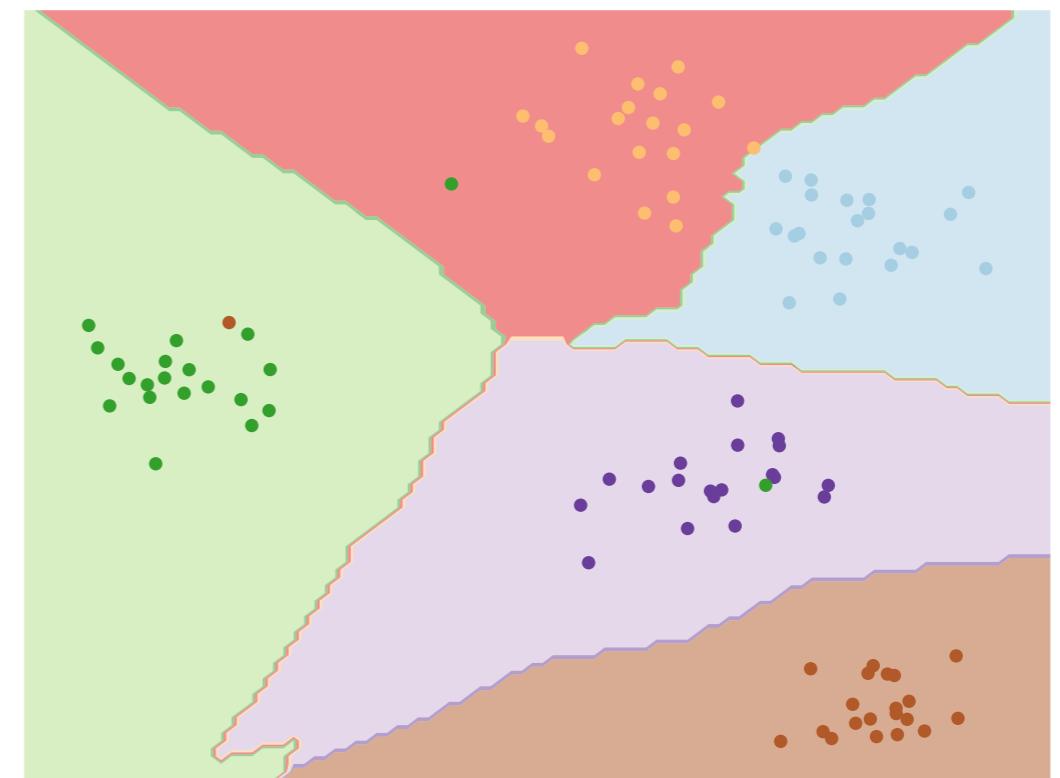
- ❖ K-Nearest Neighbors : Vamos a clasificar nuestro ejemplo según la clasificación de la mayoría de los K más parecidos

Frontera de decisión

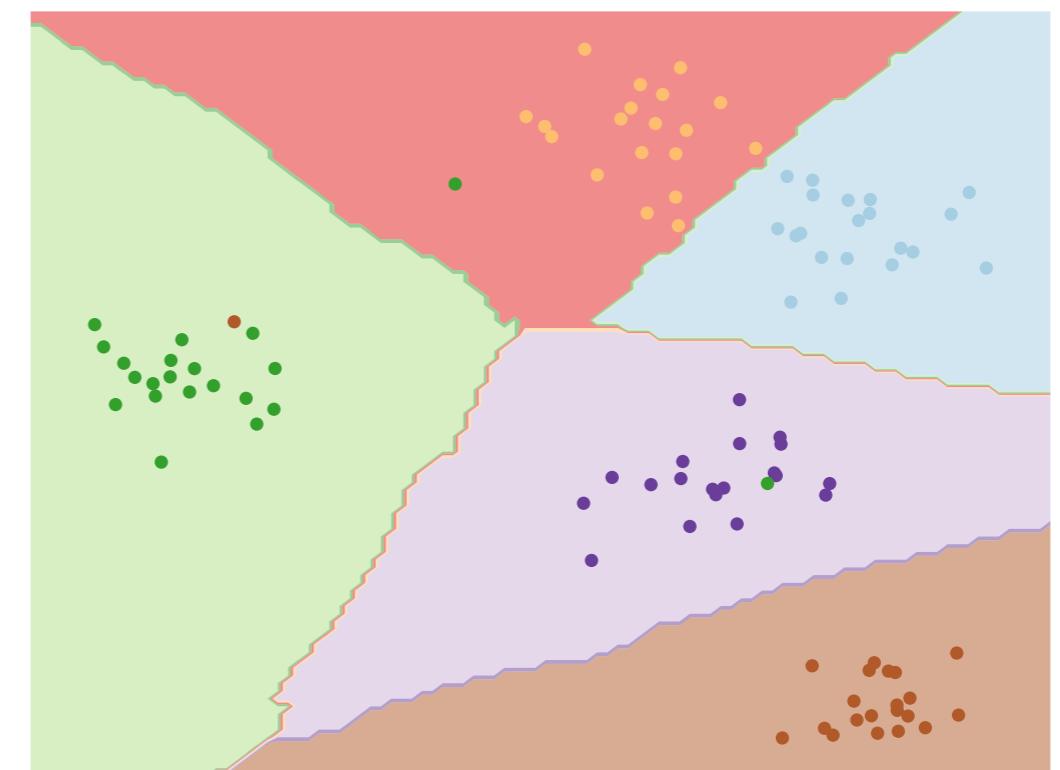
❖ K-Nearest Neighbors



K = 1



K = 3



K = 7

Hiperparámetros

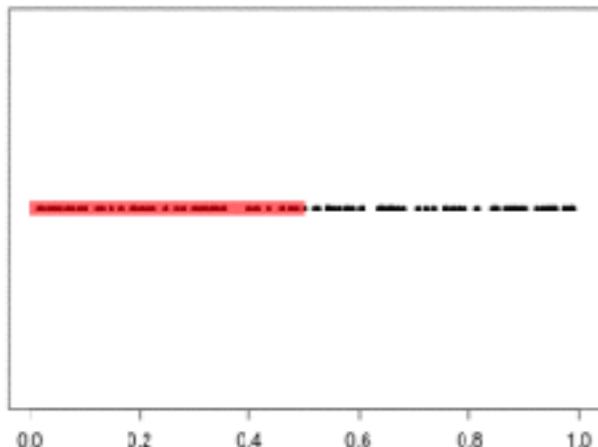
- ❖ ¿Qué métrica utilizar (L1, L2, etc.) ?
- ❖ ¿Cuántos vecinos utilizar ?
 - ¡¡ Completamente dependiente del problema !!

La maldición de la dimensionalidad

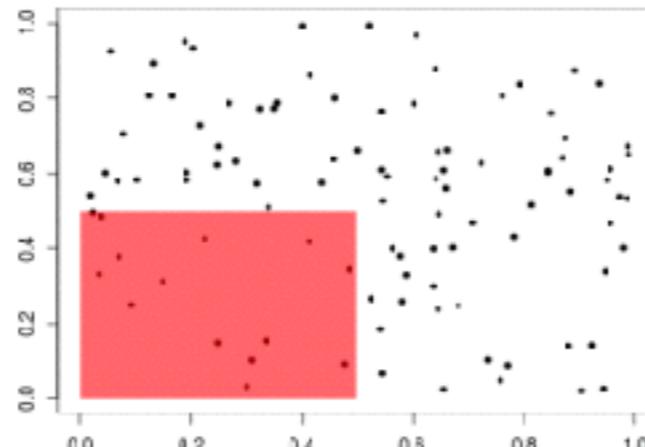
- ❖ La cantidad de muestras a utilizar no escala frente al incremento de la dimensionalidad del problema (o características).

Ej. supongamos una distribución uniforme de los posibles valores de las características

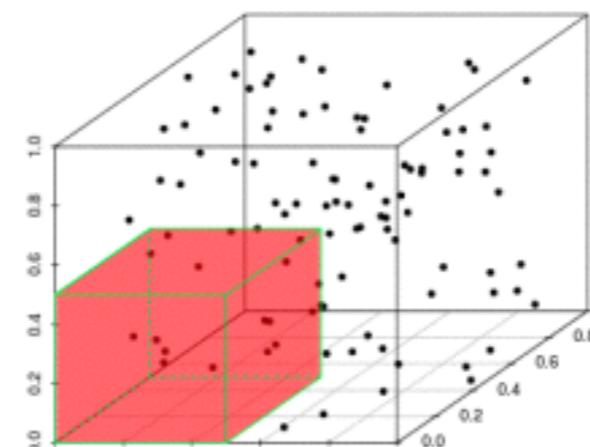
1D: **42%** capturado



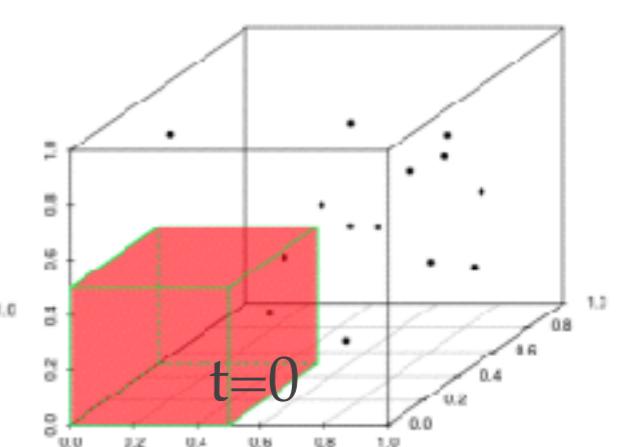
2D: **14%** capturado



3D: **14%** capturado



4D: **3%** capturado



Clasificación de imágenes: Clasificación lineal

- ❖ De forma similar a la regresión lineal, buscamos resolver un problema del estilo:

$$\mathbf{y} = W\mathbf{x}$$

dónde \mathbf{y} representa algún tipo de distancia de la imagen \mathbf{x} a cada una de las 10 clases posibles.

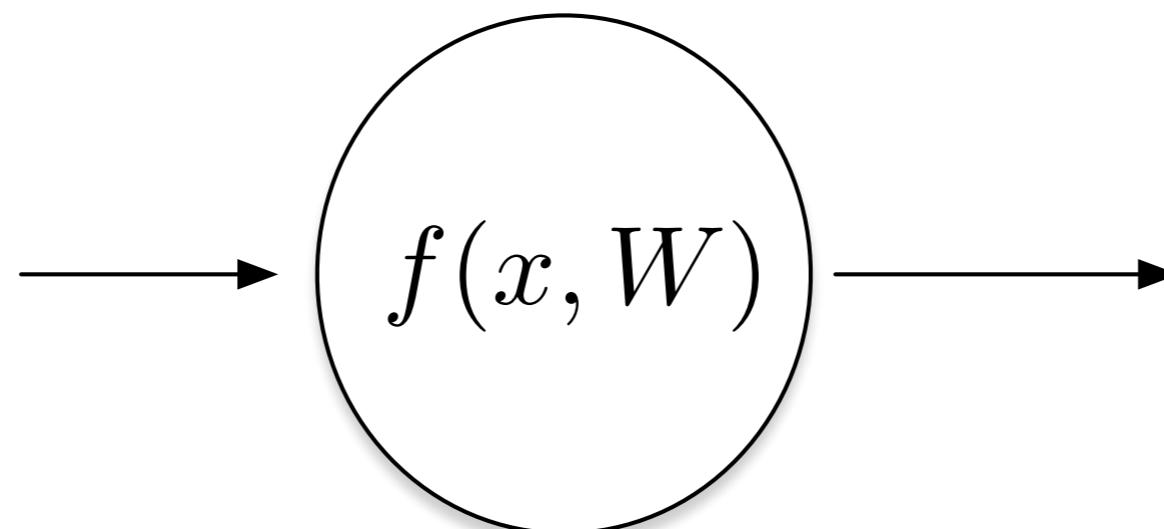
Clasificación lineal

$$f(x, W) = Wx = \hat{W}x + b \quad \xrightarrow{\text{bias}} \quad x_0 = 1$$



x

32x32x3
(#3072)

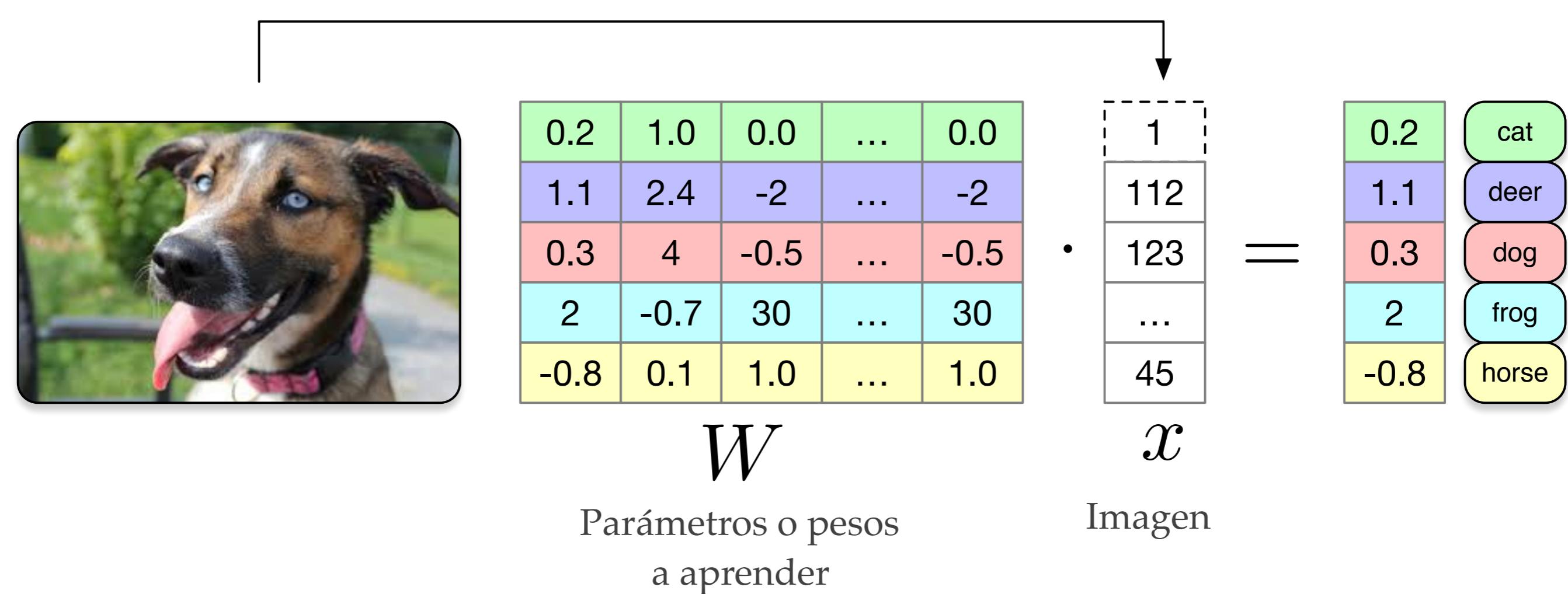


-9.7	plane
2.1	car
0.2	bird
1.1	cat
-12	deer
30.8	dog
15.0	frog
2.3	horse
10.1	ship
0.2	truck

10x1

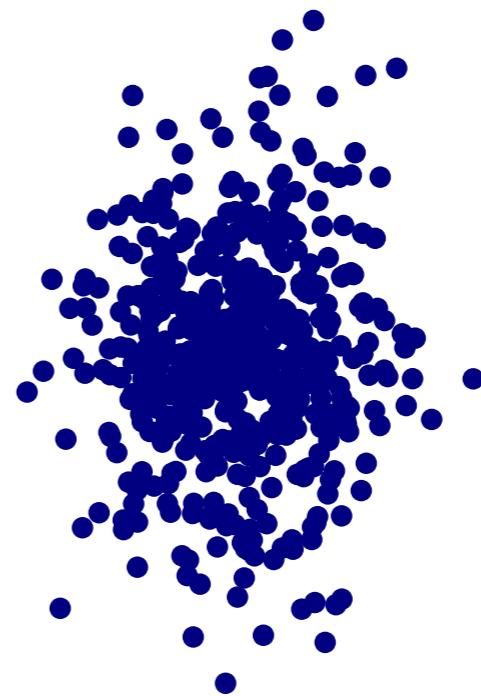
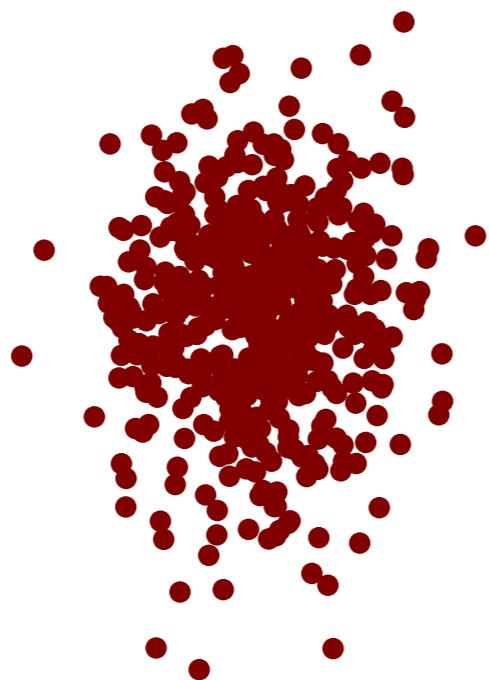
Clasificación lineal

$$f(x, W) = Wx = \hat{W}x + b \quad \xrightarrow{\text{bias}} \quad x_0 = 1$$

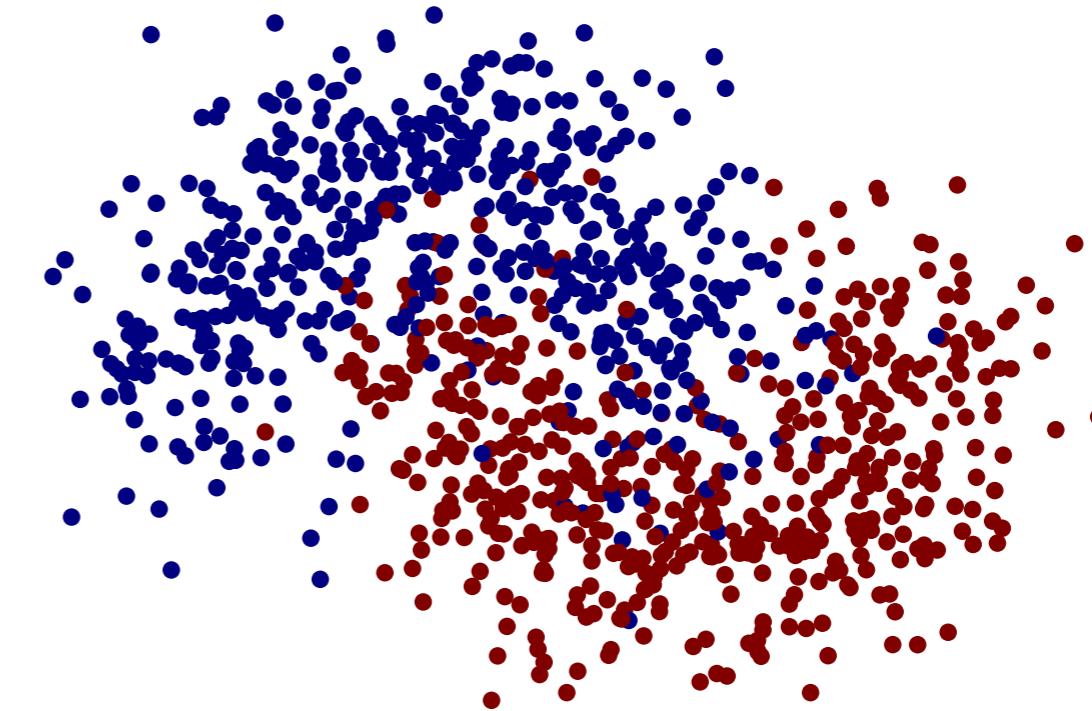
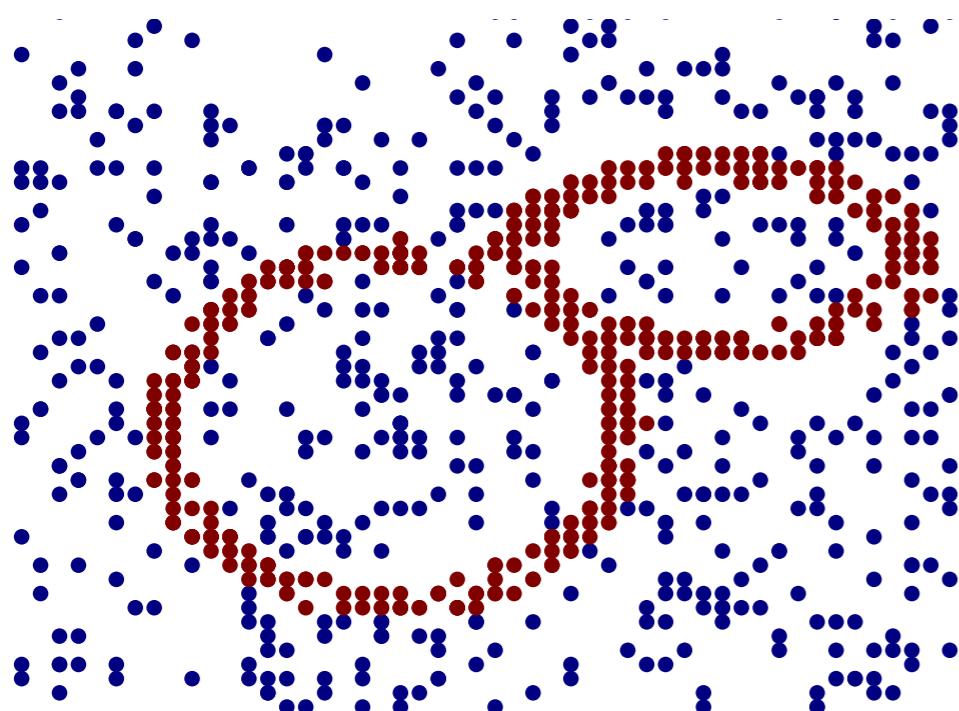
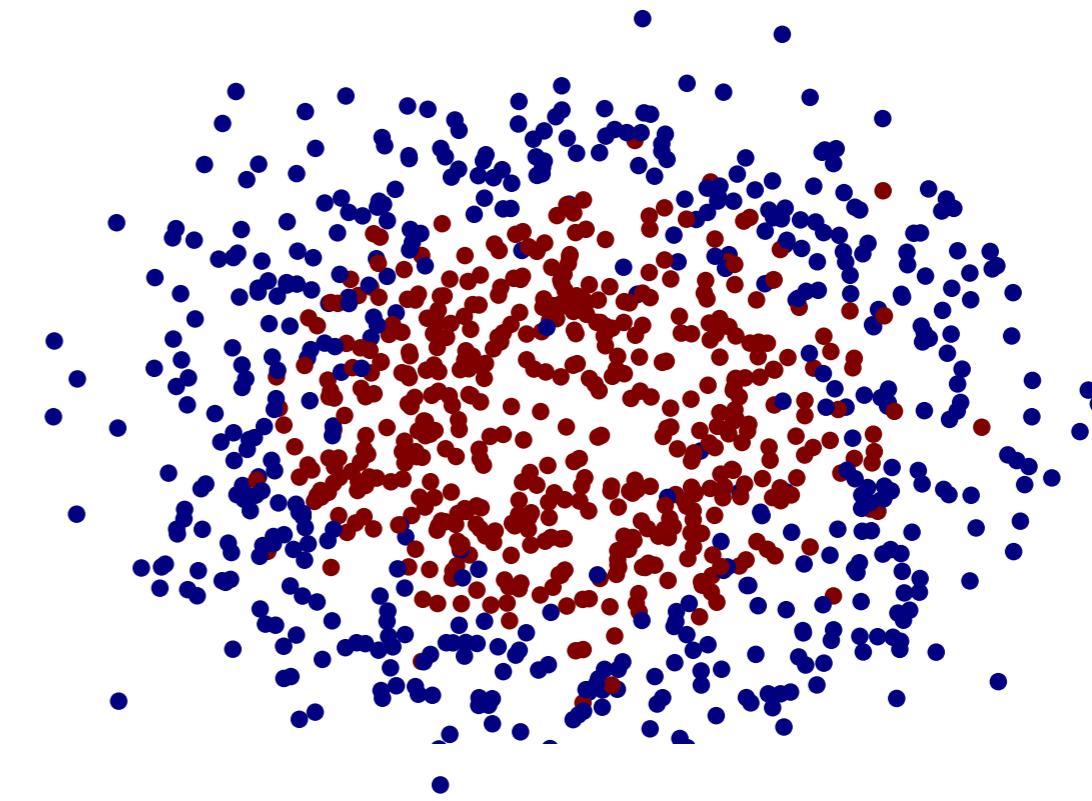
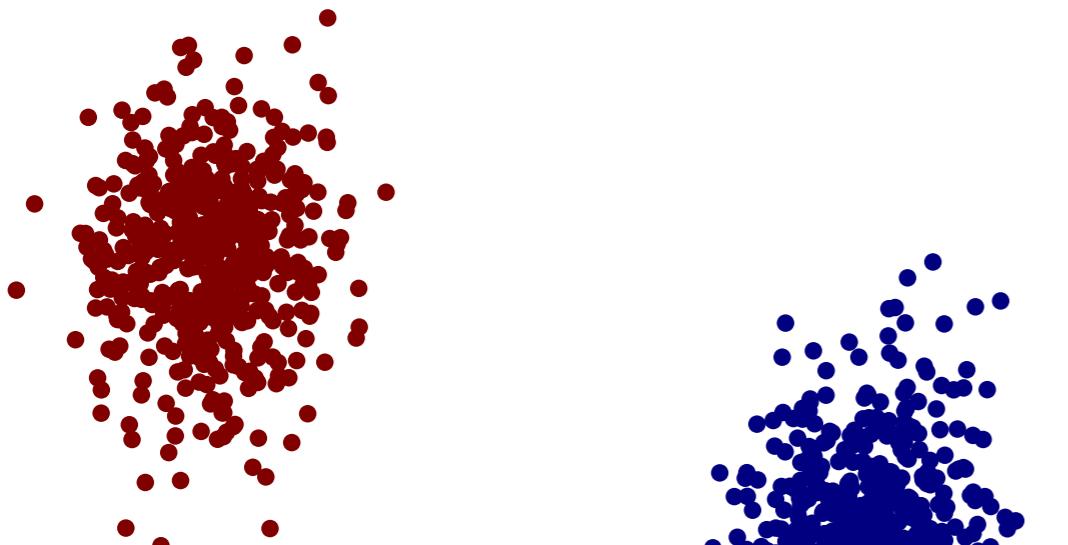


Clasificación

- ❖ ¿Qué tienen en común los métodos vistos: cuadrados mínimos, k-means y clasificación lineal ?



Clasificación lineal



Clasificación de imágenes

- ❖ Nearest Neighbors
- ❖ K-Nearest Neighbors
- ❖ Linear Classifier :
 - ❖ Support Vector Machine
 - ❖ Softmax Classifier

Nos queda pendiente