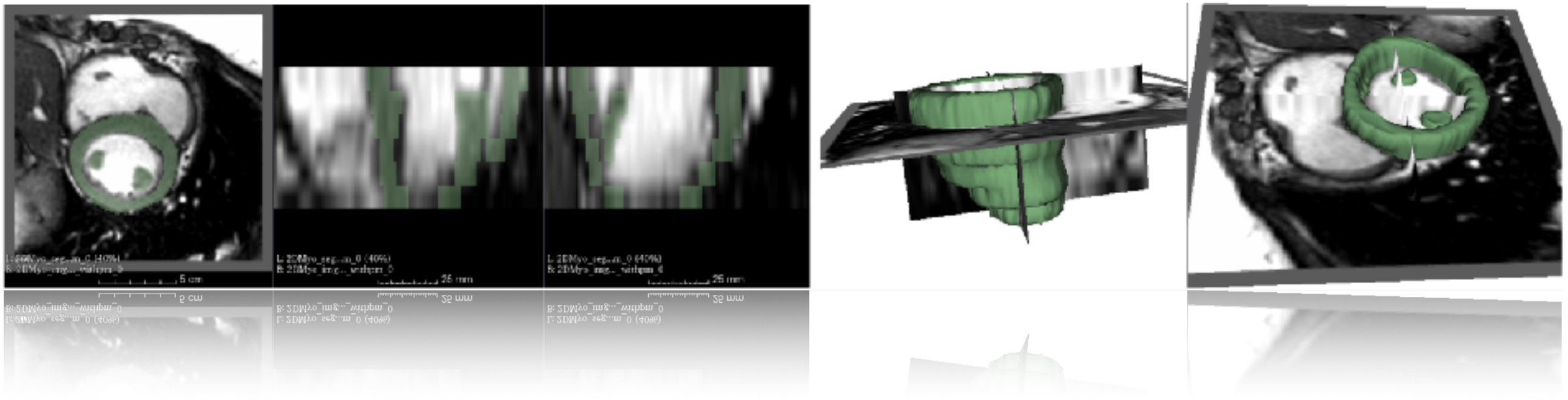


Departamento de Física Médica - Centro atómico Bariloche - IB

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.

CONICET

UNCUYO
UNIVERSIDAD NACIONAL DE CUYO

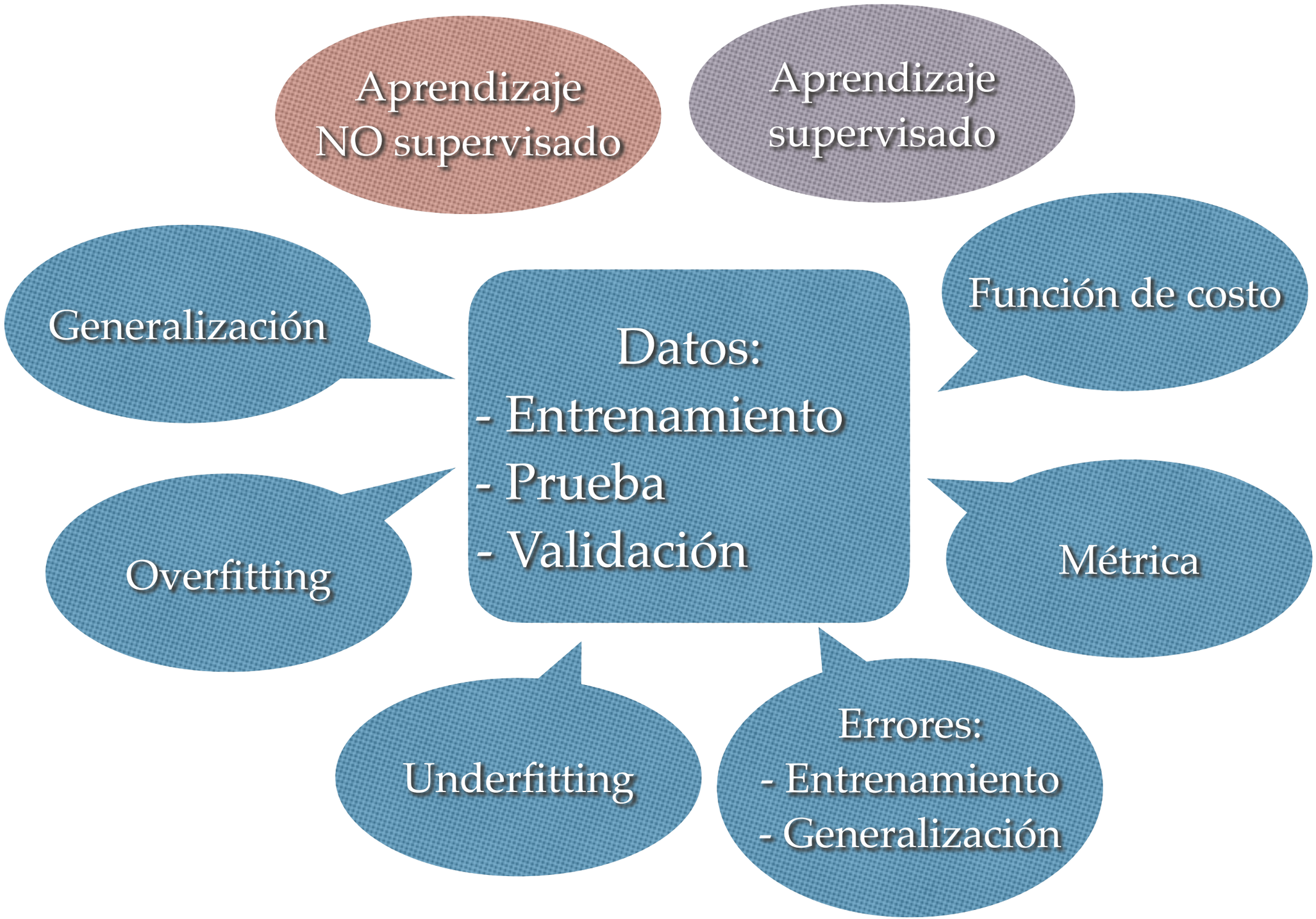
Instituto Balseiro

CNEA

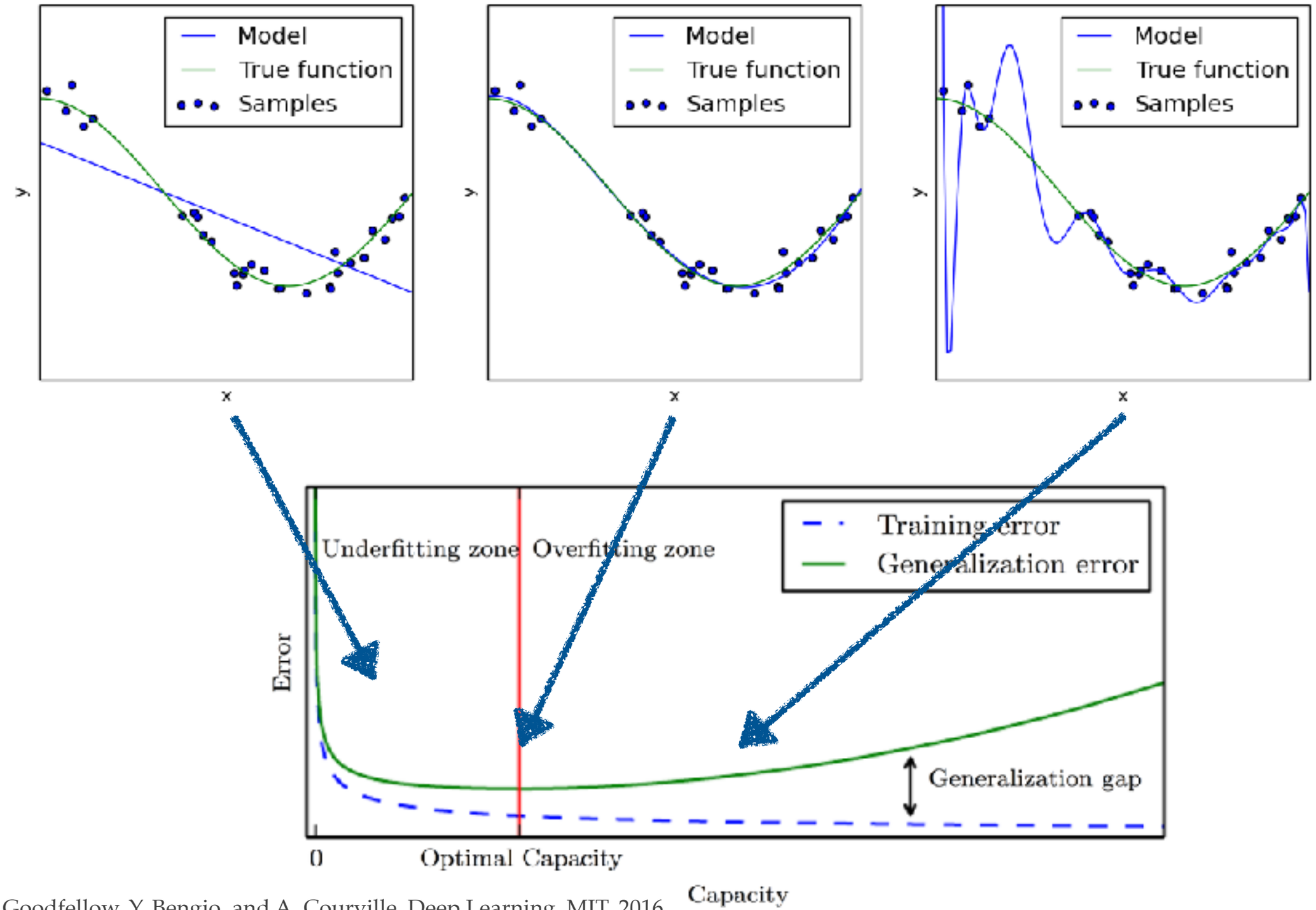
Machine Learning

- 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



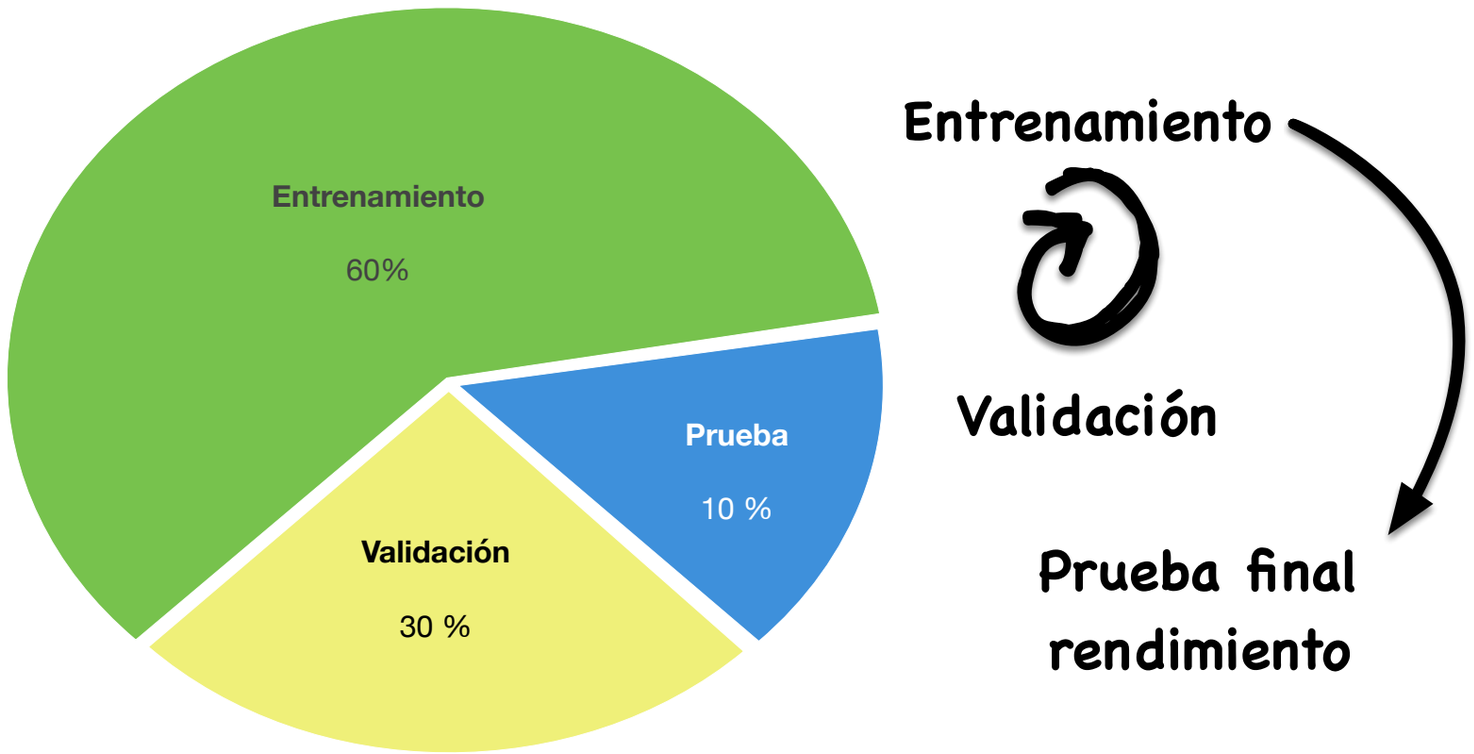
Underfitting - Overfitting



Img. source: I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT, 2016.

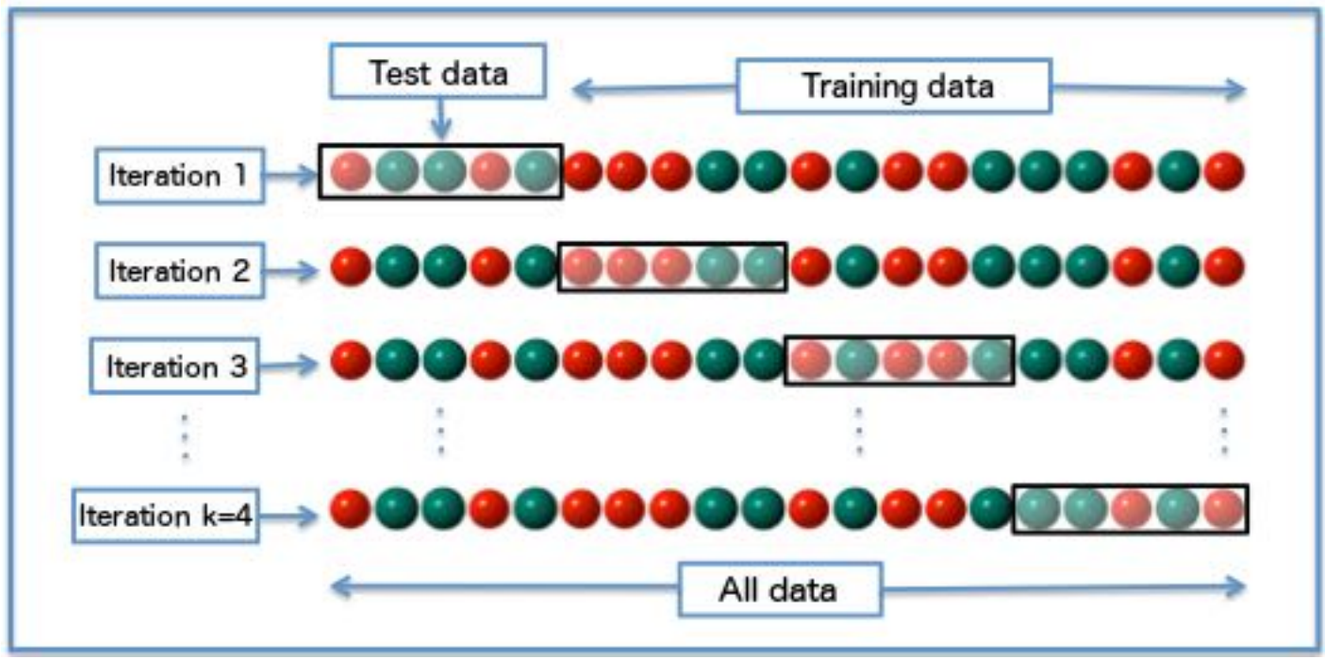
Generalización

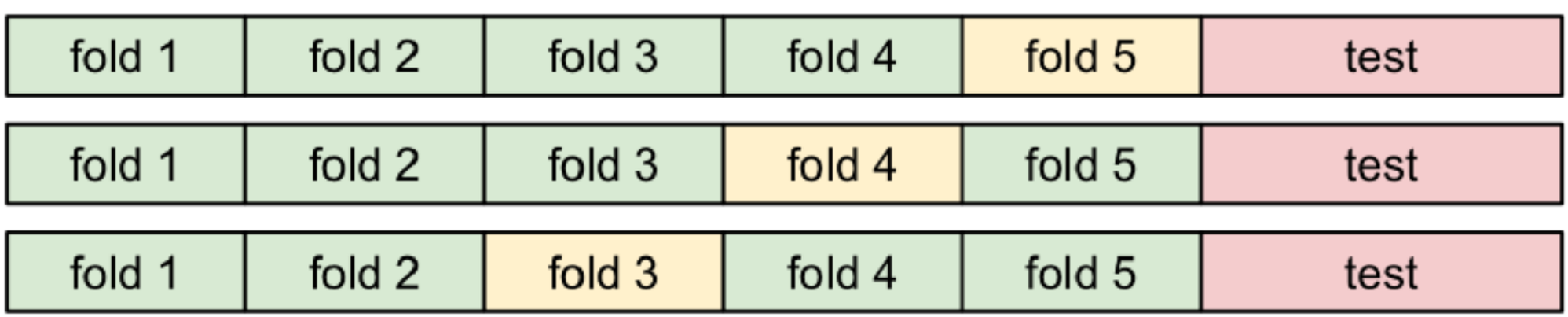
- Nuestro objetivo es Generalizar el conocimiento.
  - Los datos de prueba no se usan en el entrenamiento, de ninguna forma, ni siquiera para determinar los **hiperparámetros**.
  - Para evaluar los hiperparámetro usamos un subconjunto de los datos de entrenamiento que llamamos datos de validación.
-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



Validación

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





Img. source: Wikipedia and Stanford CS231n 2017

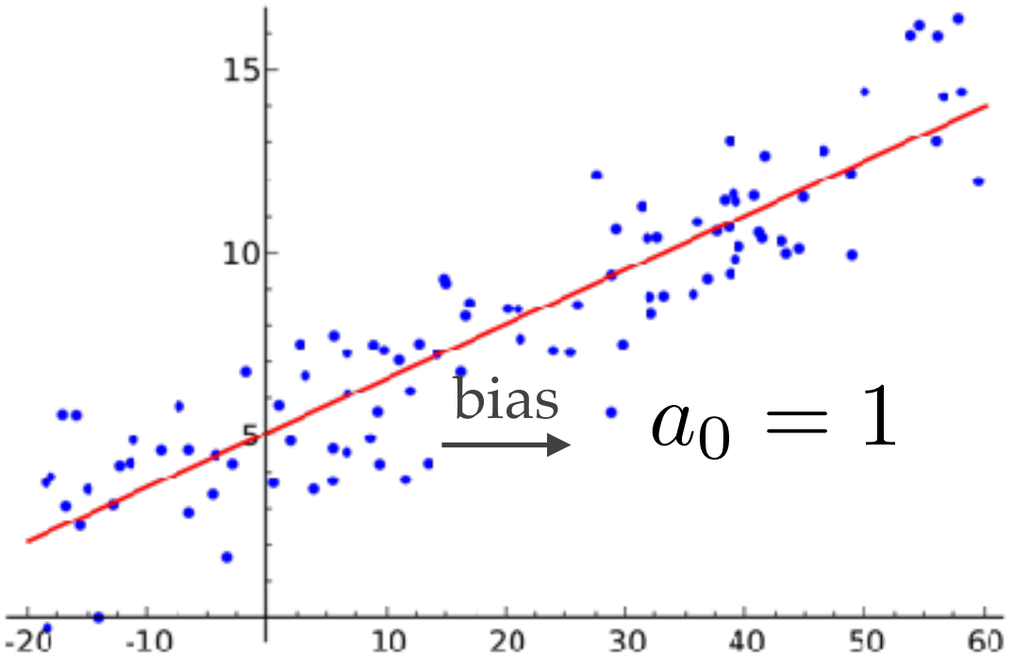
Supervisado y No Supervisado

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)

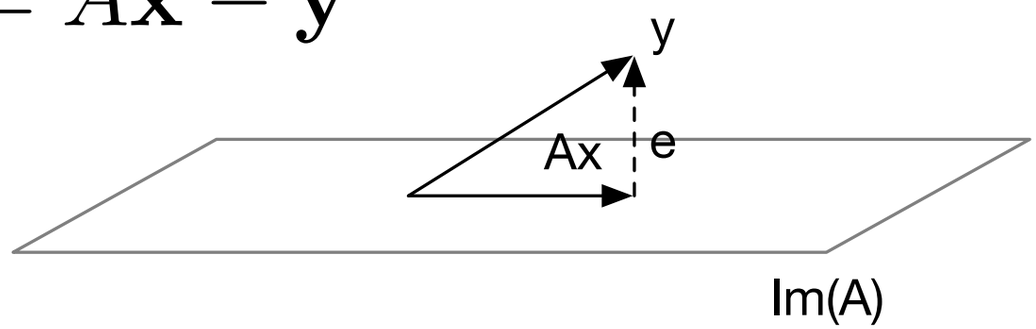
Objetivo: predecir el valor $y \in \mathbb{R}$ dado un conjunto de características $\mathbf{a} \in \mathbb{R}^n$ mediante una relación lineal:



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

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

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

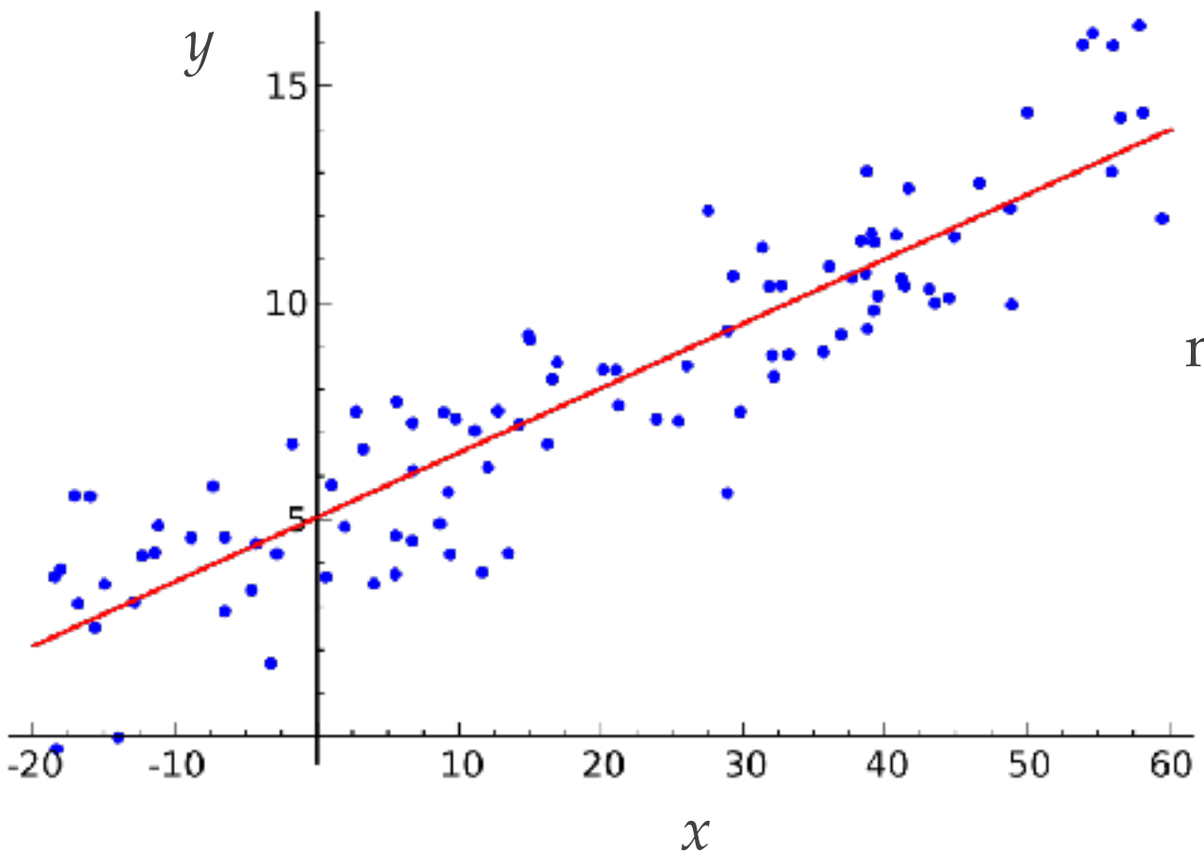


$$\min ||\mathbf{e}|| \Rightarrow \mathbf{e} \perp A\mathbf{s} \Rightarrow < \mathbf{e}, A\mathbf{s} > = \mathbf{s}^T A^T (A\mathbf{x} - \mathbf{y}) = 0$$

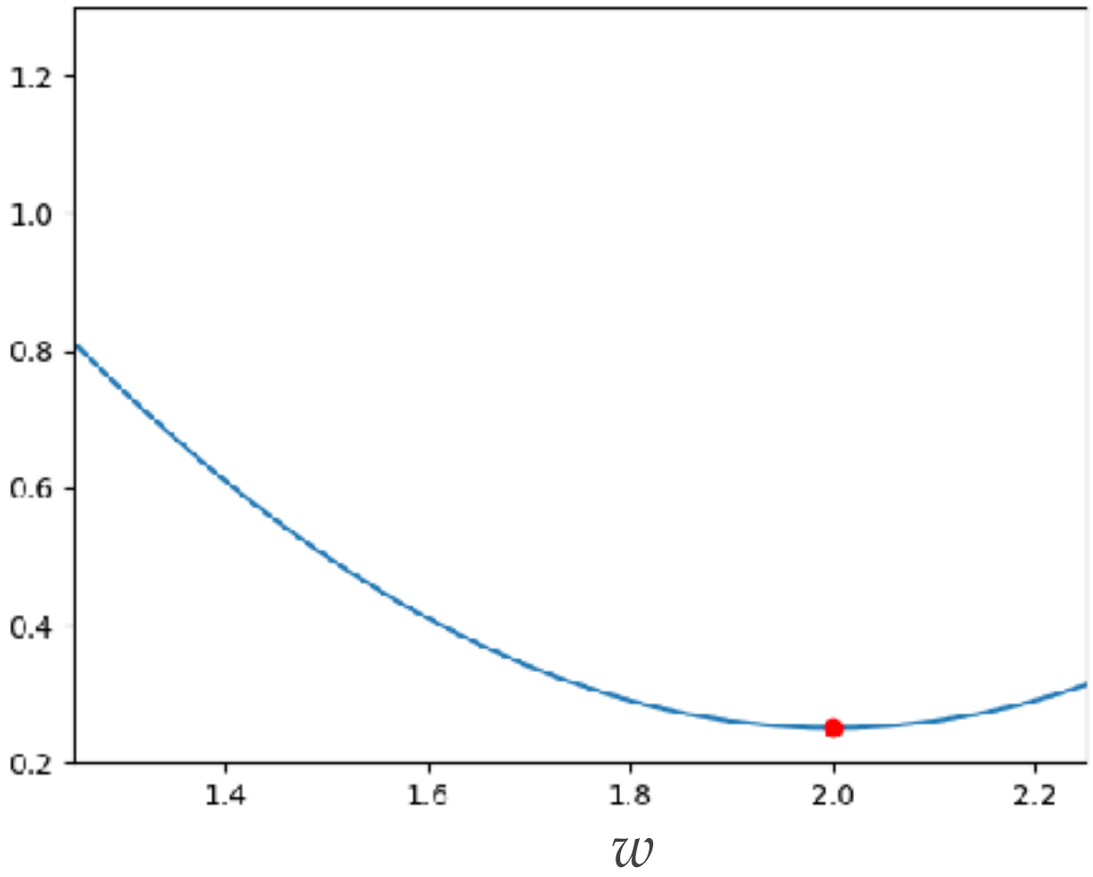
$$\boxed{\mathbf{x} = (A^T A)^{-1} A^T \mathbf{y}}$$

Regresión Lineal

¿Qué tiene de bueno esta función de costo?



mse



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

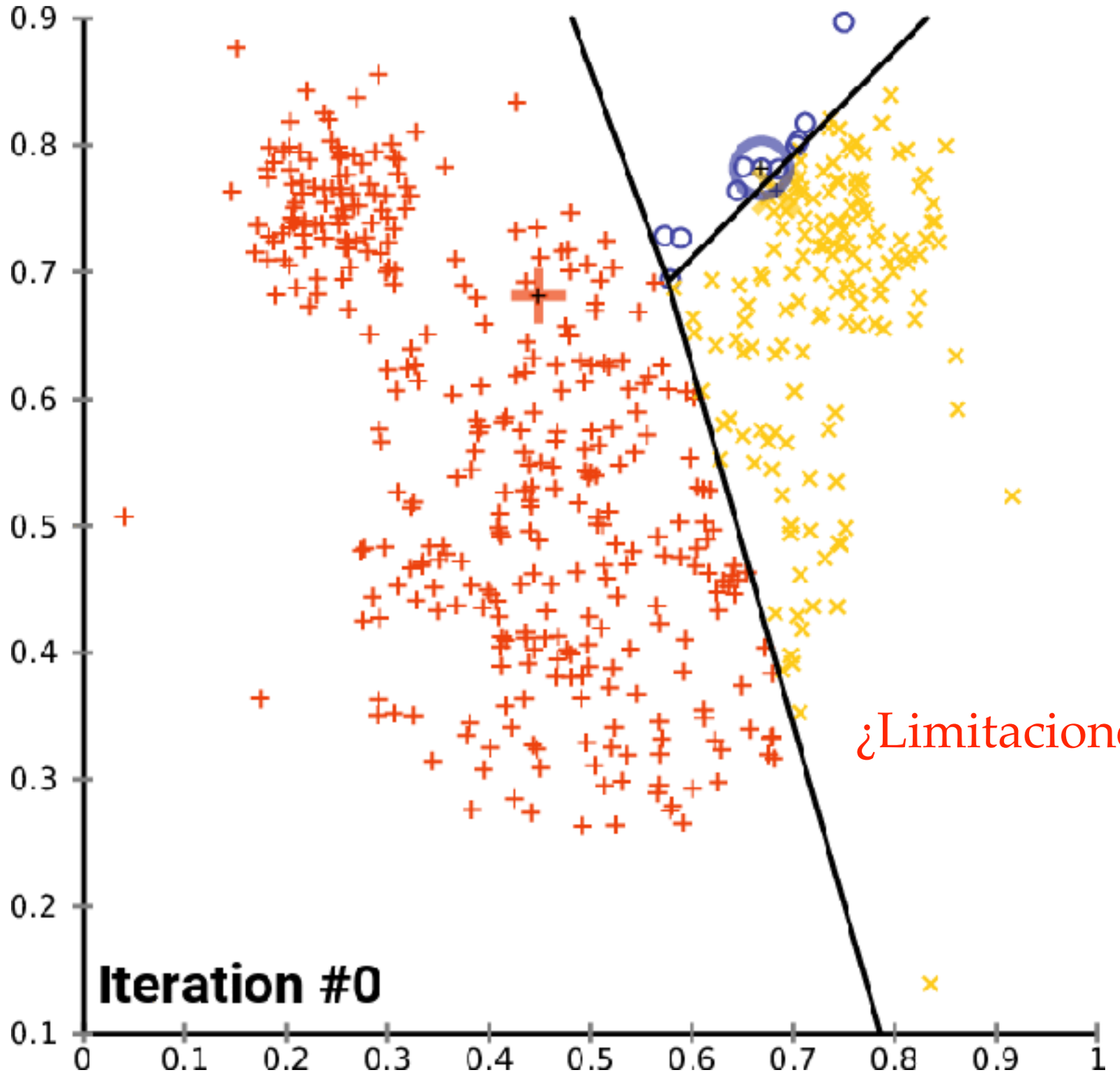
K-means (no supervisado)

- 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 $\mu_1, \ldots, \mu_k$
  - 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

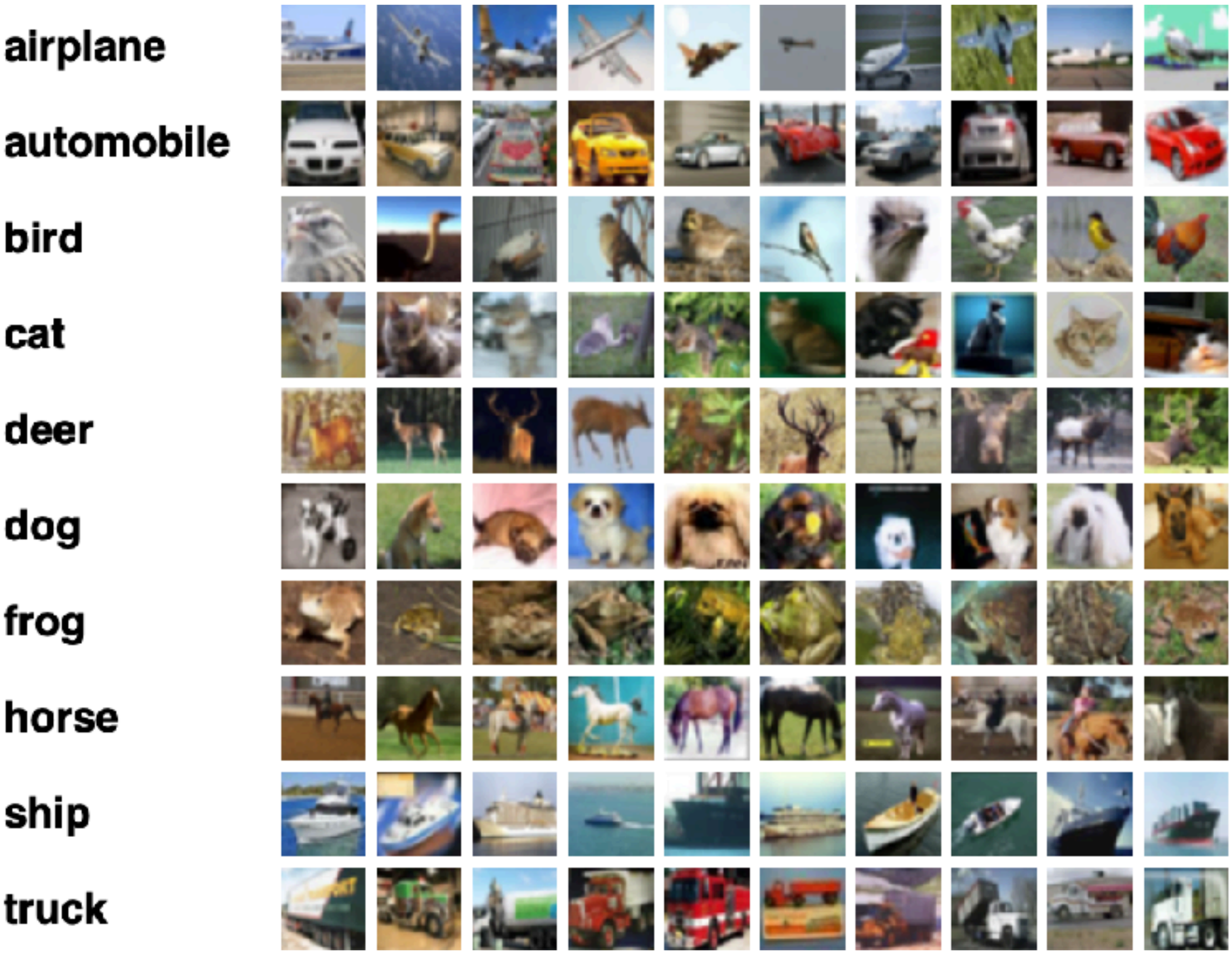
K-means



¿Limitaciones del método?

Clasificación de imágenes

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

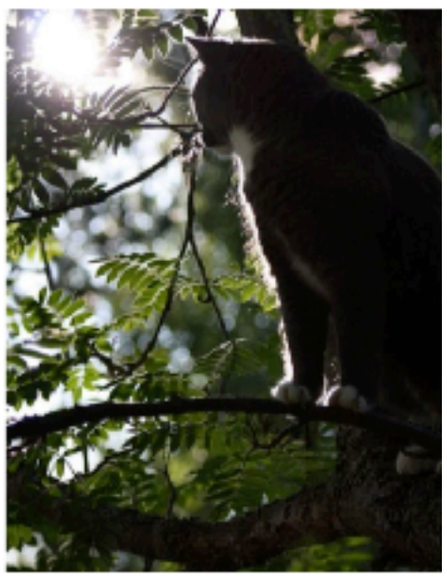


Desafíos

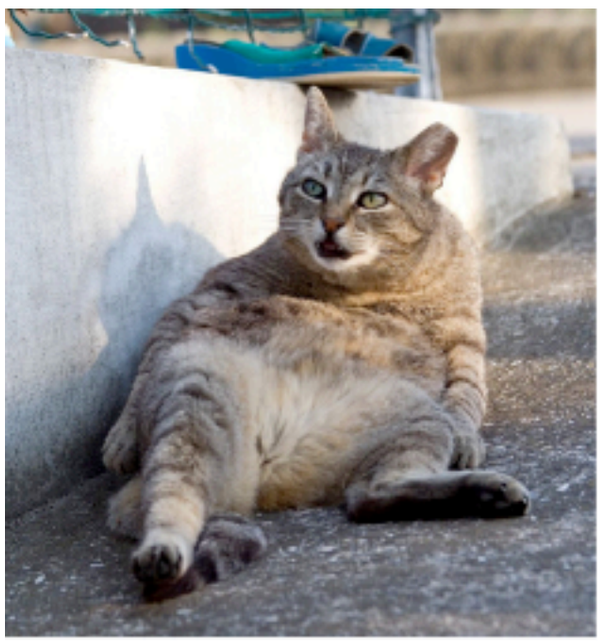
This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

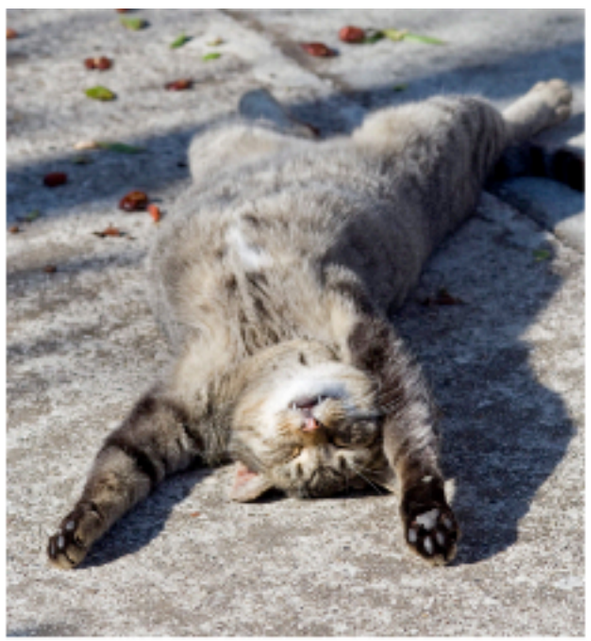
This image is CC0 1.0 public domain



This image is CC0 1.0 public domain

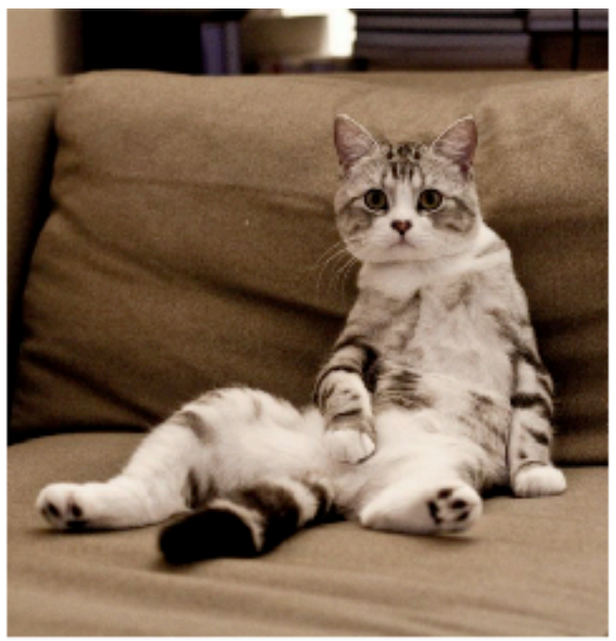


This image by Umberto Salvagnin is licensed under CC-BY 2.0



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 Thai is licensed under CC-BY 2.0

Img. source: Stanford CS231n 2017

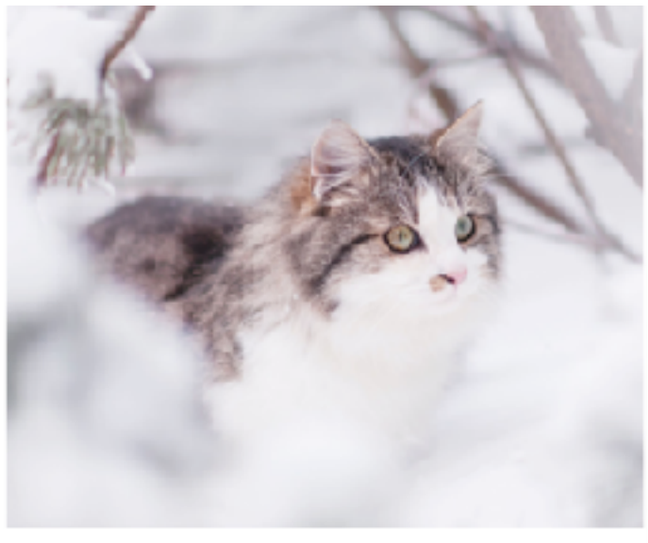
Desafíos

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

Img. source: Stanford CS231n 2017

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 las misma clasificación del ejemplo de entrenamiento cuya similitud es la máxima

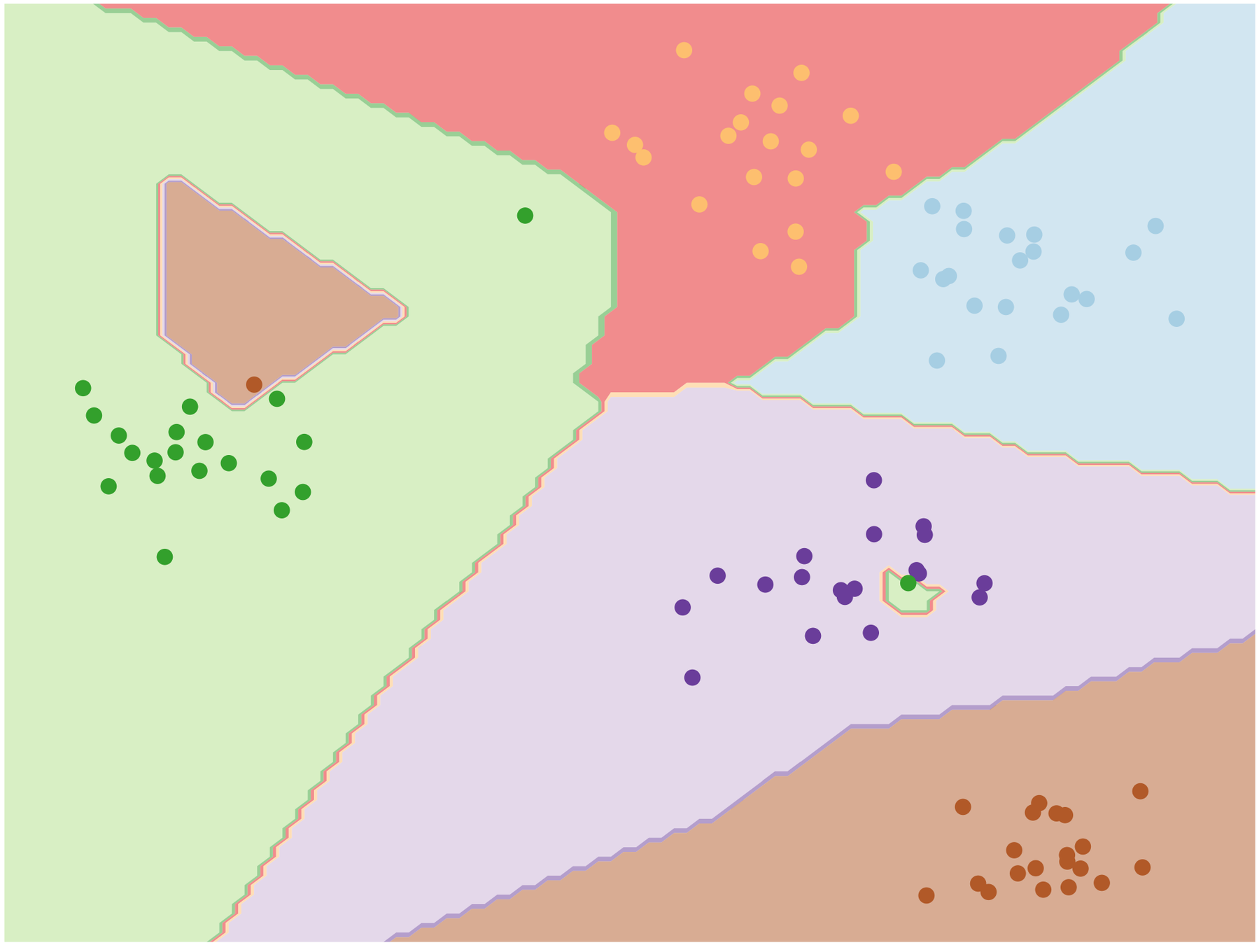
```python
from keras.datasets import cifar10
import numpy as np



class NearesNeighbor:

    def __init__(self):
        self.X = None
        self.Y = None

    def train(self, X, Y):
        # Las img vienen en (n, ror,col,rgb)
        self.im_shape = X.shape[1:]
        self.X = np.reshape(X, (X.shape[0], np.prod(self.im_shape)))
        self.Y = Y

    def predict(self, X):
        assert self.X is not None, 'Train method needs to be call first'
        Yp = np.zeros(X.shape[0], np.uint8)
        for idx in range(X.shape[0]):
            norm = np.linalg.norm(self.X - X[idx].ravel(), axis=-1)
            idmin = np.argmin(norm)
            Yp[idx] = self.Y[idmin]

        return Yp


#  num_classes = 10

# The data, split between train and test sets:
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')


model = NearesNeighbor()
model.train(x_train, y_train)
yp = model.predict(x_test[:10])
```

Almacenamos los datos

Buscamos el más parecido

Cargamos los datos con keras

Frontera de decisión

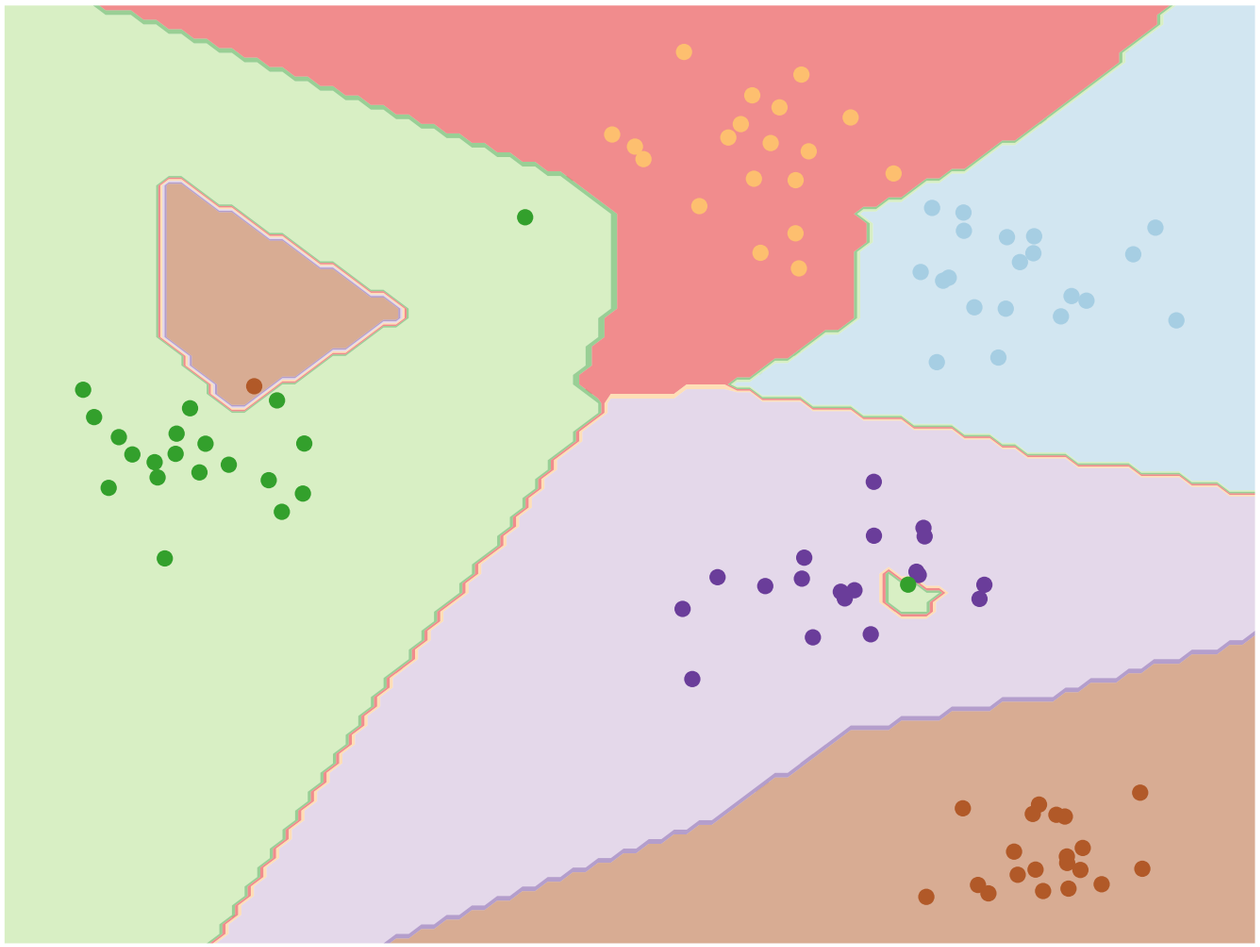


Clasificación de imágenes: KNN

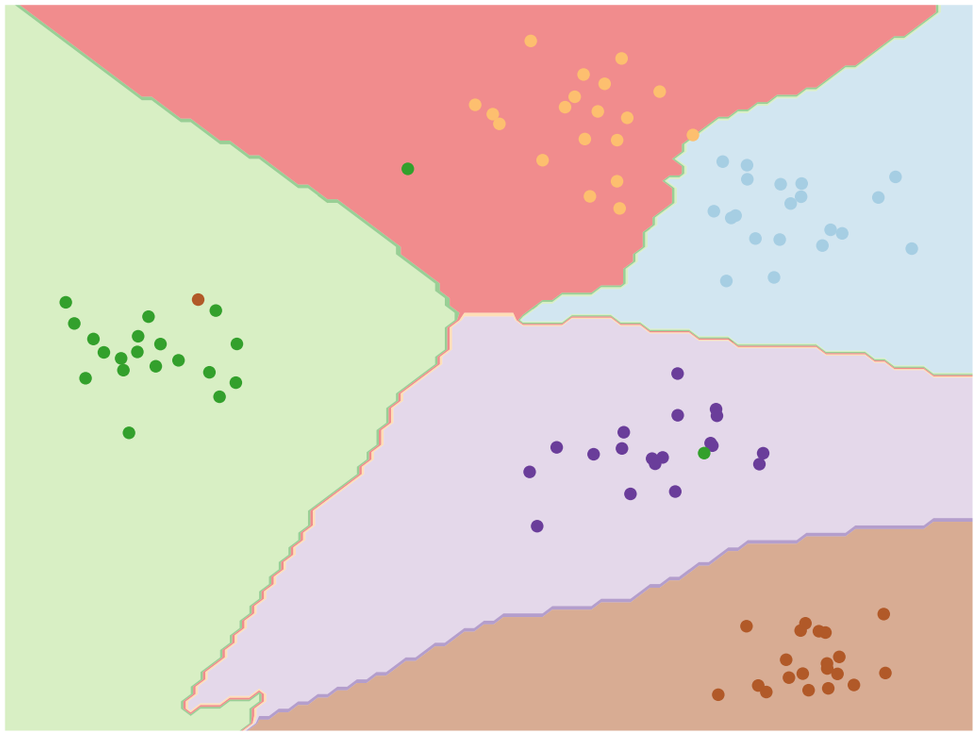
- K-Nearest Neighbors : Vamos a clasificar nuestro ejemplo según la clasificación de la mayoría de los K más parecidos

Frontera de decisión

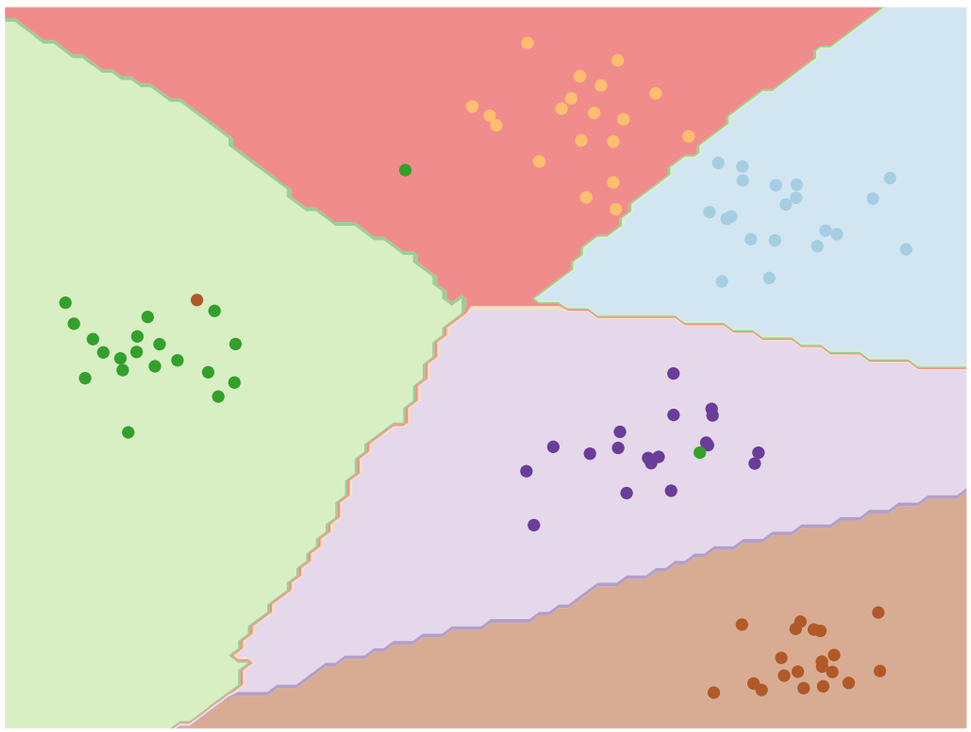
- K-Nearest Neighbors



K = 1



K = 3



K = 7

Hiperparámetros

- ¿Qué métrica utilizar (L1, L2, etc.) ?
- ¿Cuántos vecinos utilizar ?

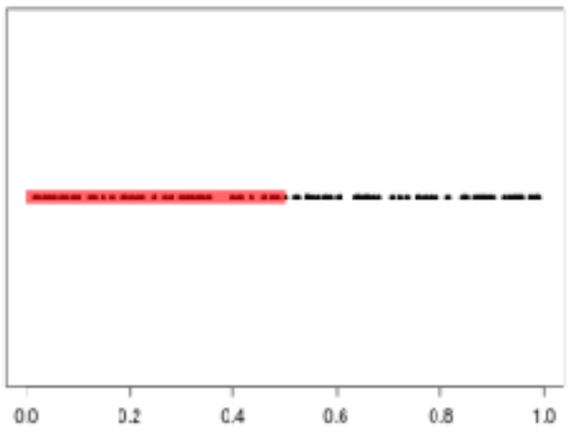
¡¡ Completamente dependiente del problema !!

La maldición de la dimensionalidad

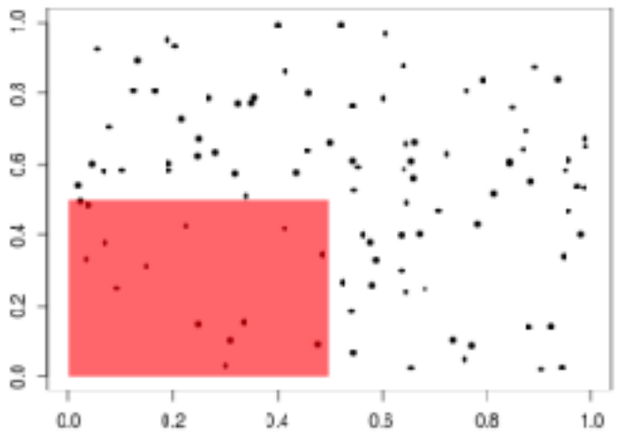
- La cantidad de muestras a utilizar no escala frente al incremento de la dimensionalidad del problema (o características).

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

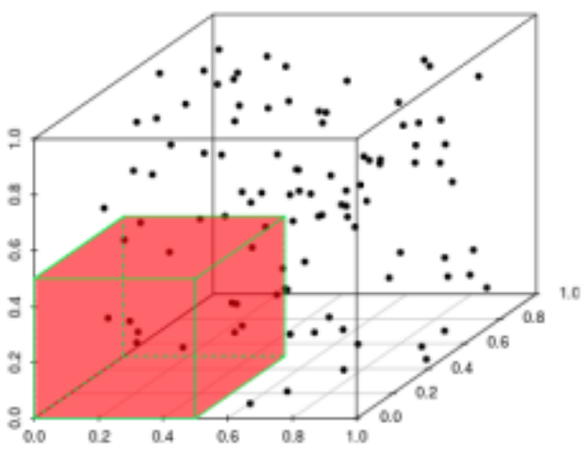
1D: 42% capturado



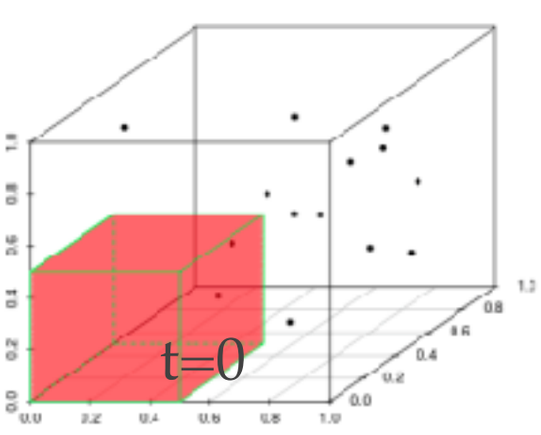
2D: 22% capturado



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

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

x

32x32x3
(#3072)

$f(x, W)$

-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

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

0.2	1.0	0.0	…	0.0
1.1	2.4	-2	…	-2
0.3	4	-0.5	…	-0.5
2	-0.7	30	…	30
-0.8	0.1	1.0	…	1.0

W

Parámetros o pesos
a aprender

$\cdot$

1
112
123
…
45

x

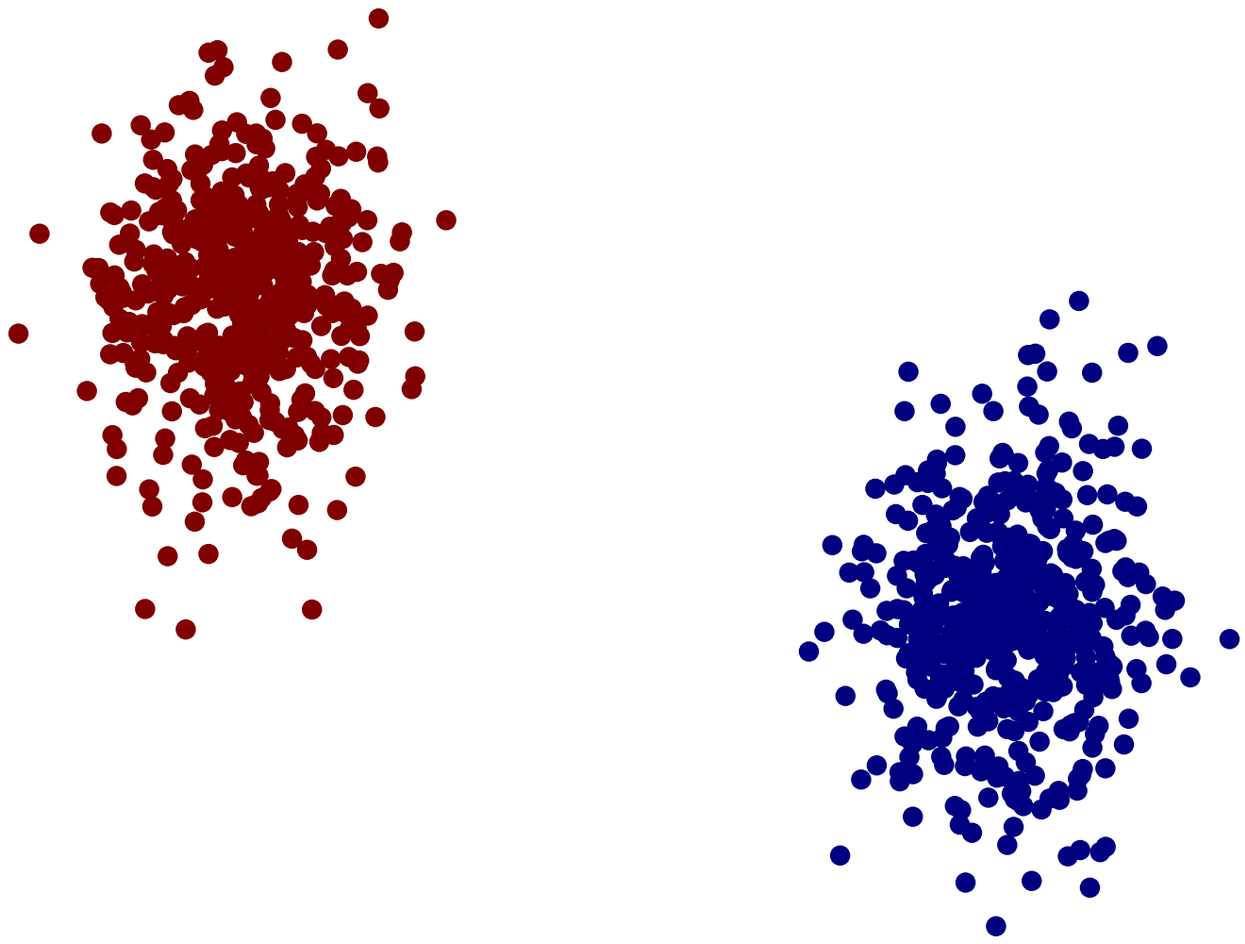
Imagen

$=$

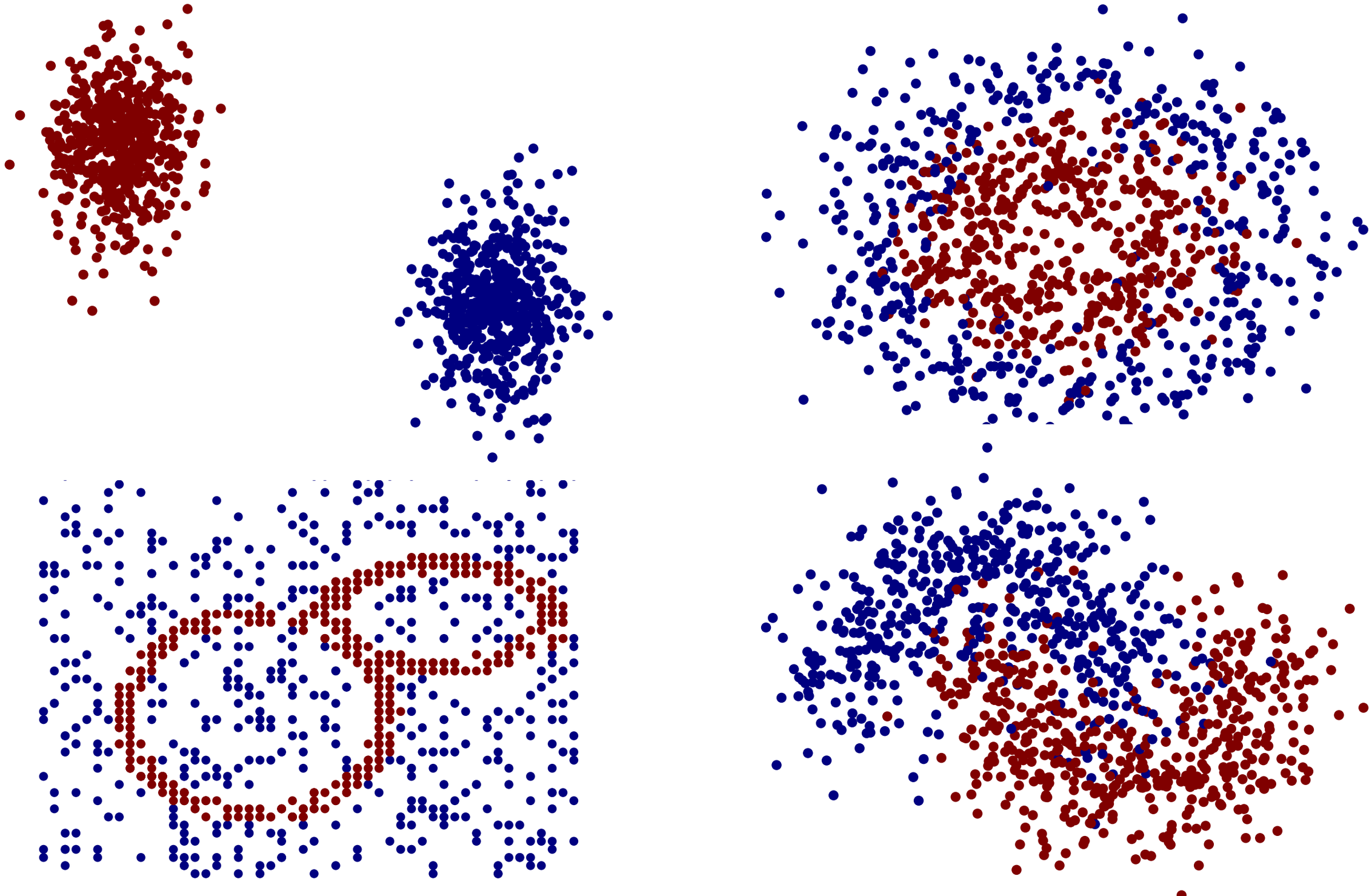
0.2	cat
1.1	deer
0.3	dog
2	frog
-0.8	horse

Clasificación

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



Clasificación lineal



Clasificación de imágenes

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

Nos queda pendiente