Machine Learning 101

Fundamentos de Machine Learning

Felipe Alonso Atienza

Data Scientist @BBVA

Índice

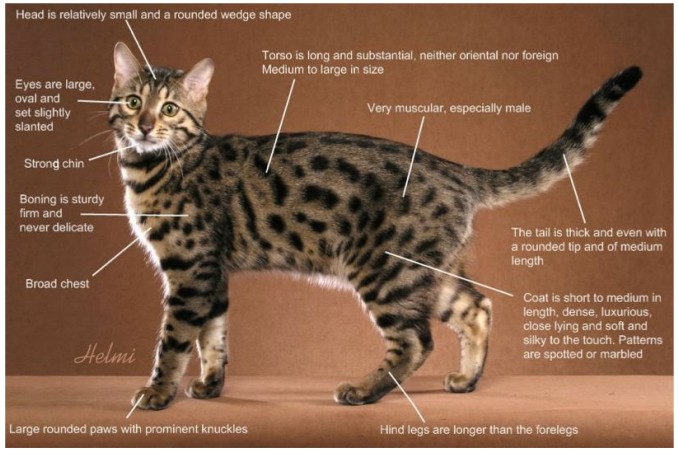
¿Qué es *machine learning*?

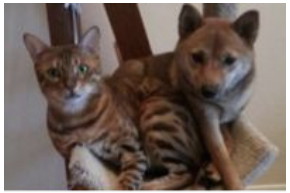
El arte y la ciencia de

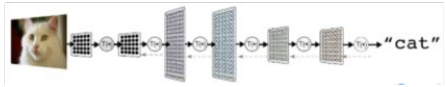
- "*Proporcionar a los ordenadores la capacidad de **aprender** a tomar decisiones a partir de los **datos**, sin ser programados explícitamente para ello*" Arthur Samuel, 1959

- Útil cuando no se puede utilizar una fórmula que describa la realidad, pero sí dispones de datos para construir una solución empírica

¿Qué es *machine learning*?

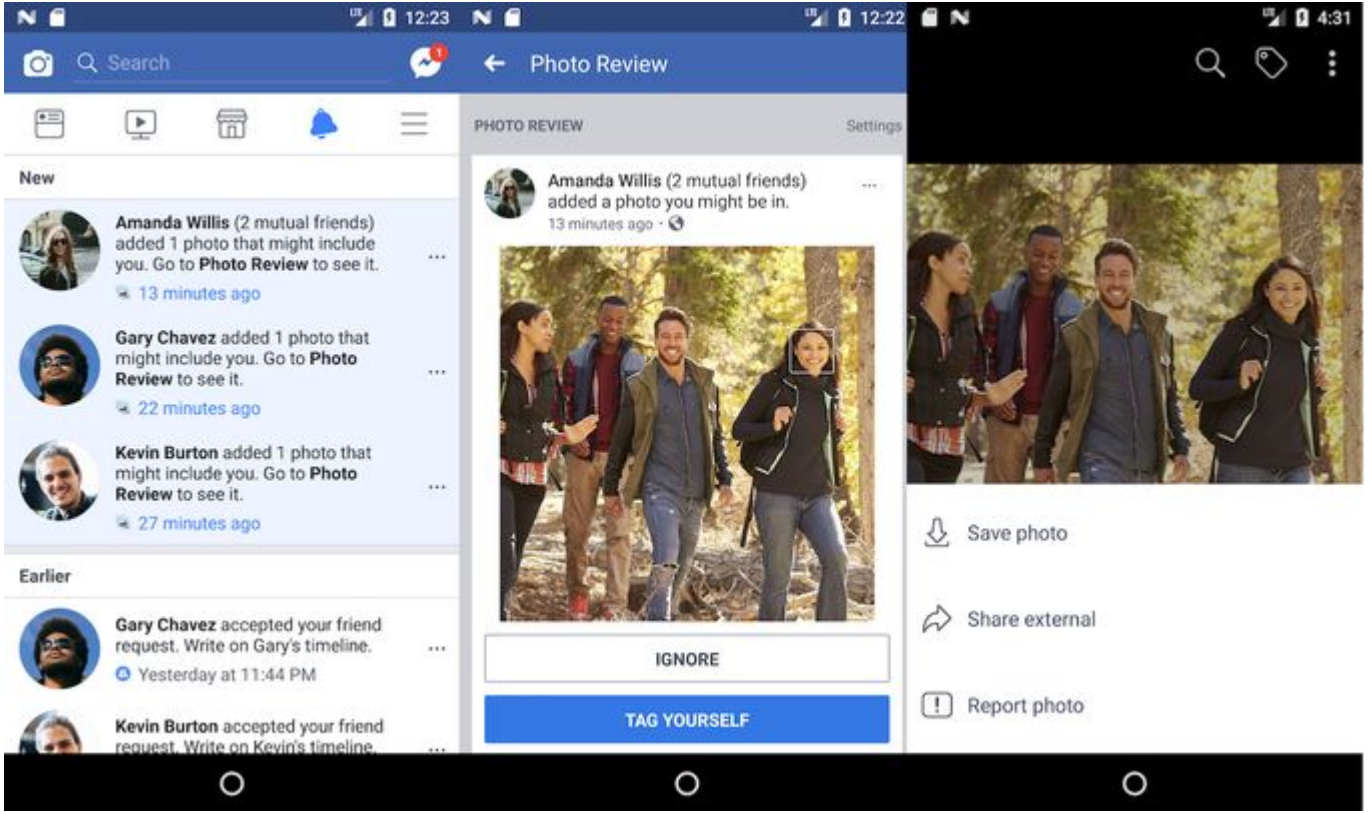






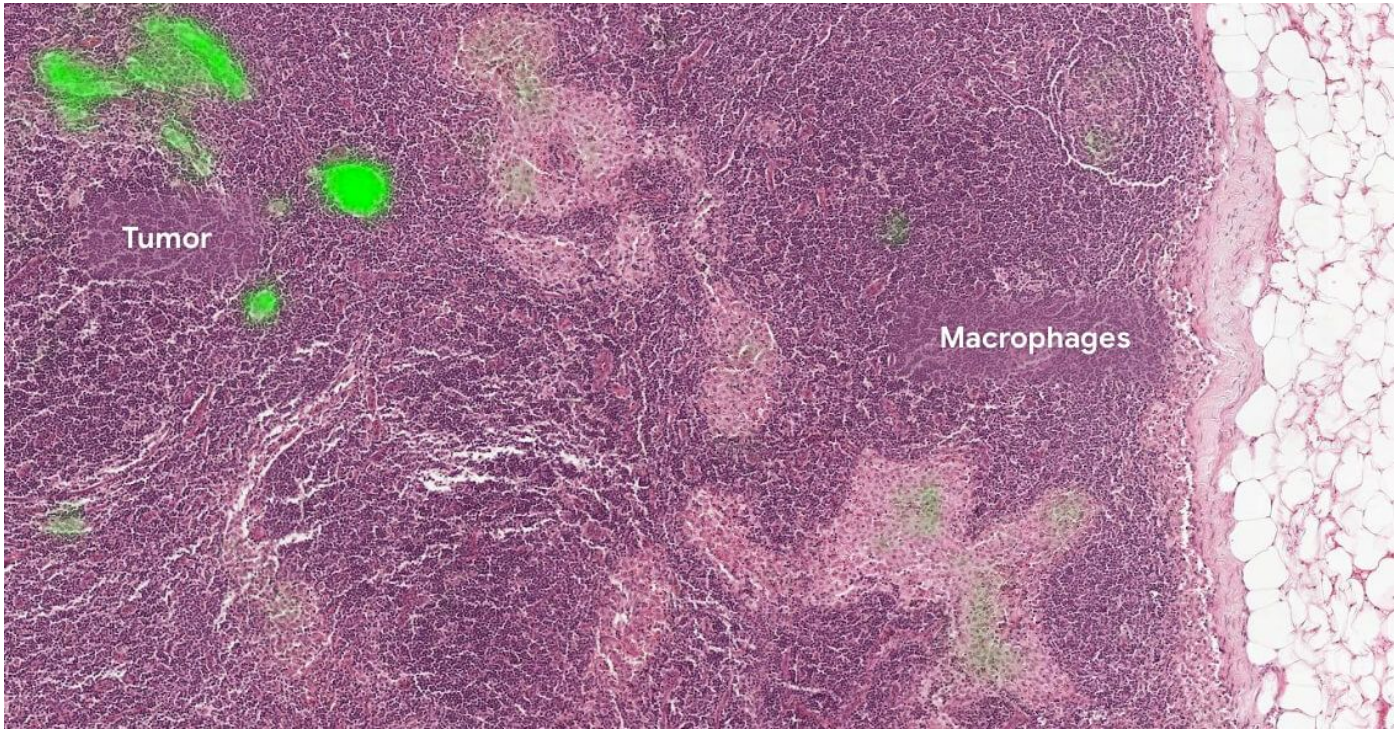
https://medium.com/@karpathy/software-2-0-a64152b37c35

¿Qué es *machine learning*?



¿Qué es *machine learning*?

¿Qué es *machine learning*?

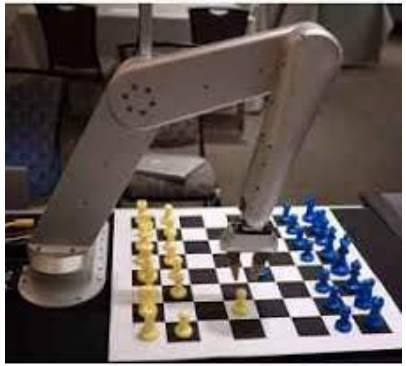


https://ai.google/research/teams/brain/healthcare-biosciences

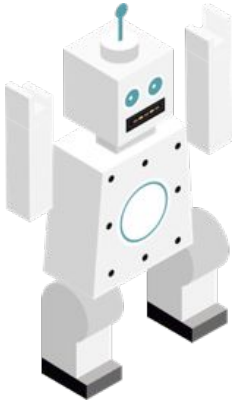
Diferencias con Inteligencia Artificial

- Inteligencia Artificial: “Programa de computación diseñado para realizar determinadas operaciones que se consideran propias de la inteligencia humana”

Narrow



General



Superintelligence

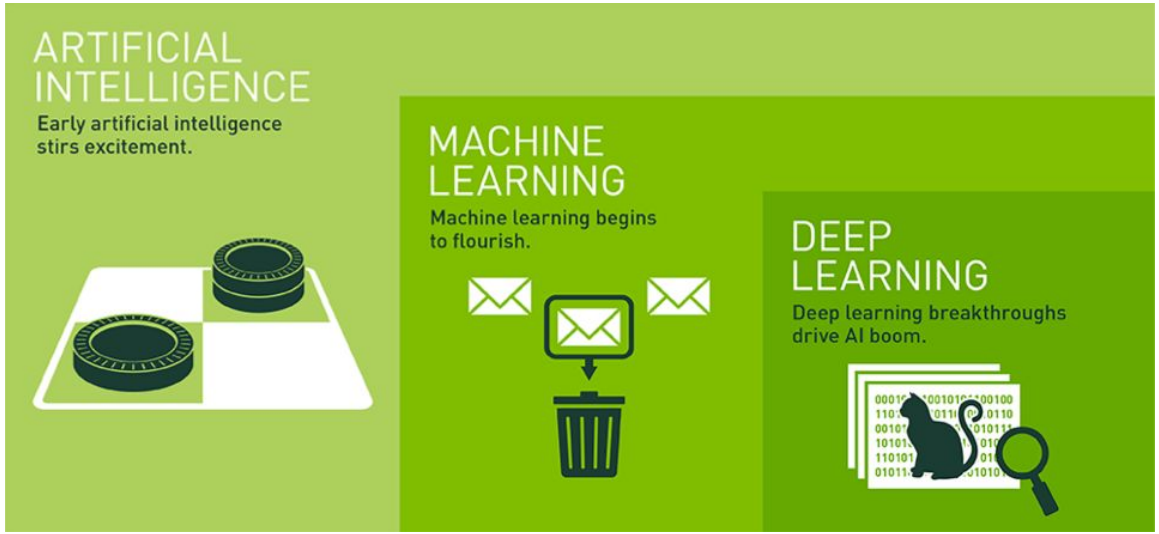
https://bdtechtalks.com/2017/05/12/what-is-narrow-general-and-super-artificial-intelligence/

Diferencias con *Deep Learning*

- Redes neuronales (algoritmo de *machine learning*)
- Arquitecturas complejas (profundas)
- Caídas en el olvido y **vuelta a la gloria** gracias a GPUs y datos masivos (digitalización)

    - Grandes resultados (superior a humanos) en datos estructurados y algoritmos supervisados
        - Imagen médica
        - Gaming

AI, ML y DL



Fuente:
https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Relación entre ML y Estadística

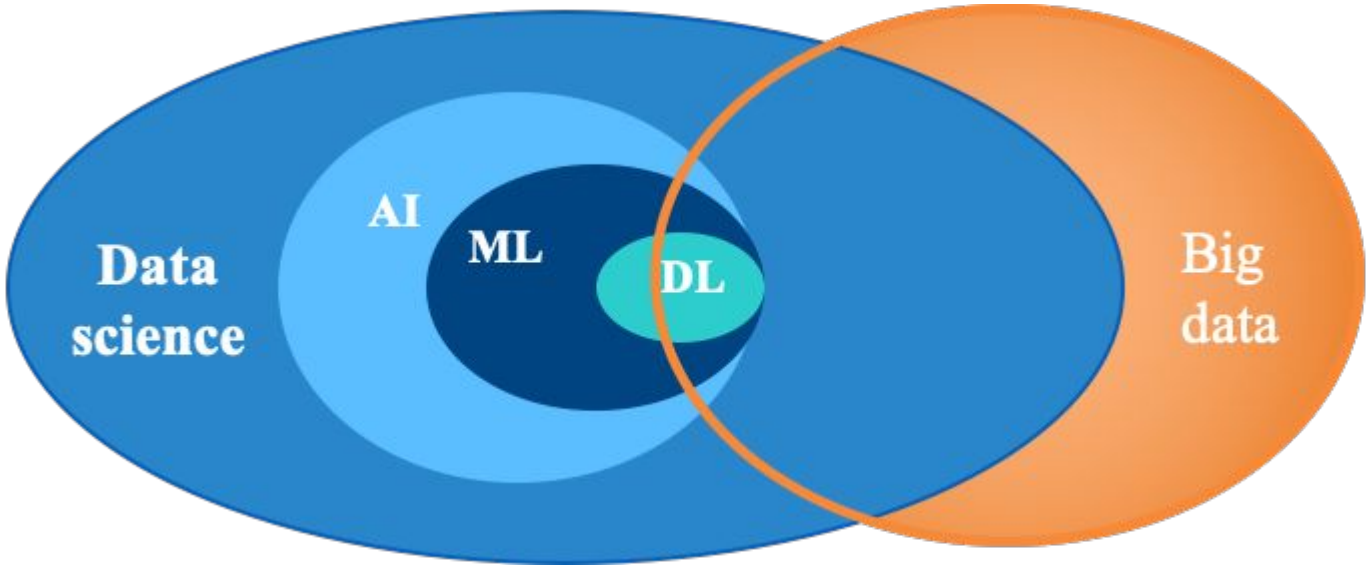
Estadística

- Modelo
- Énfasis en inferencia

Machine Learning

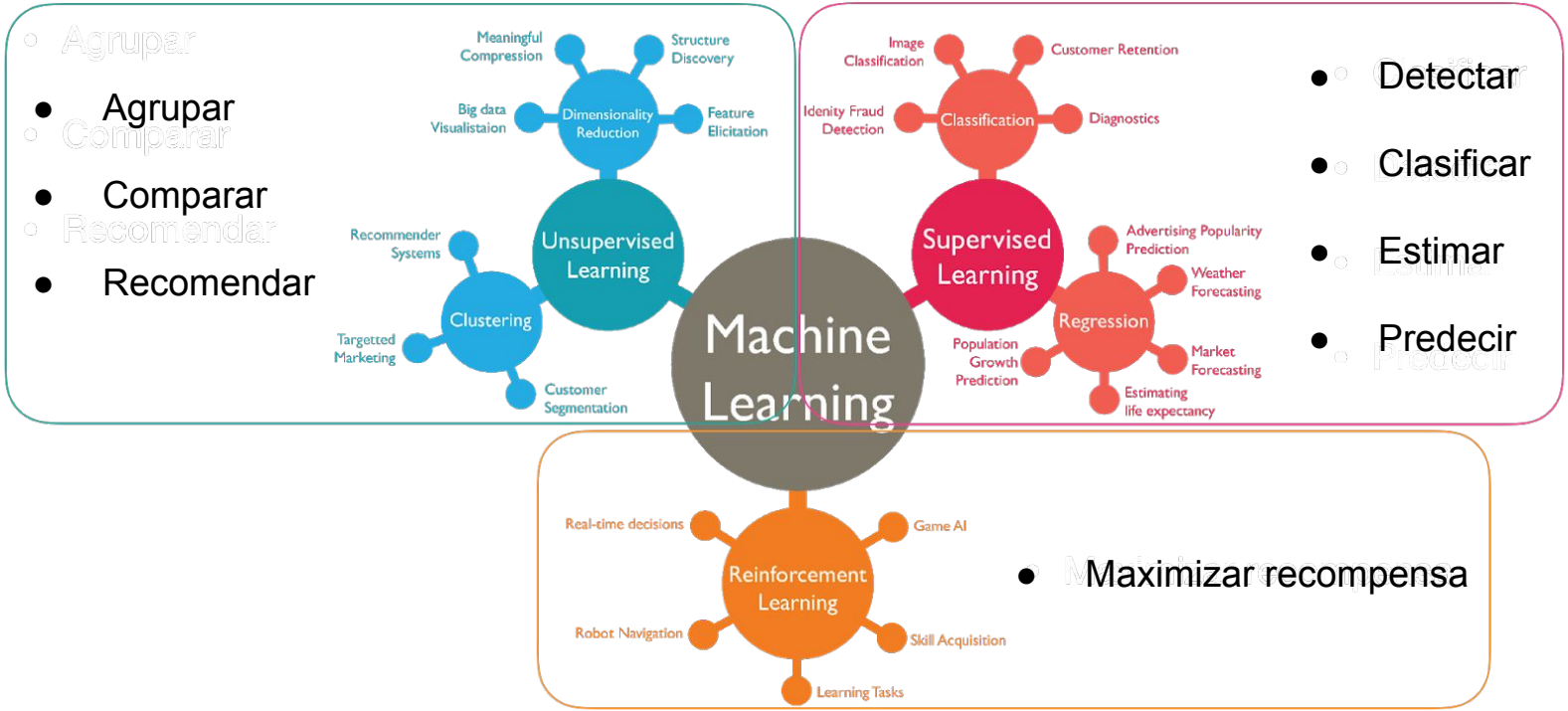
- Datos
- Predicción

Relación entre ML y ciencia de datos



Índice

Tipos de *machine learning*



https://medium.com/marketing-and-entrepreneurship/10-companies-using-machine-learning-in-cool-ways-887c25f913c3

Aprendizaje supervisado

$$\{\mathbf{x}^{(i)}, y^{(i)}\} \propto p(x, y) \text{ i.i.d.},$$

$$\mathbf{x}^{(i)} \in \mathbb{R}^d,$$

$$y^{(i)} \in \mathbb{R},$$

$$i = 1, \ldots, N,$$

$$f_{\boldsymbol{\omega}}(\mathbf{x}^{(i)}) \approx y^{(i)}$$

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.8	2.8	5.1	2.4
1	6.0	2.2	4.0	1.0
2	5.5	4.2	1.4	0.2
3	7.3	2.9	6.3	1.8
4	5.0	3.4	1.5	0.2

	Species
0	virginica
1	versicolor
2	setosa
3	virginica
4	setosa

Iris data set: https://es.wikipedia.org/wiki/Iris_flor_conjunto_de_datos

Clasificación y regresión (supervisado)

Clasificación

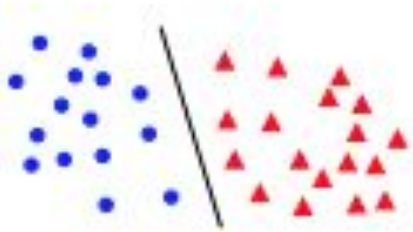
- target *y* es **discreta**
- Ej: Apto / No apto
- Regresión logística

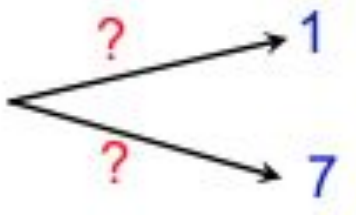
Regresión

- target *y* es **continua**
- Ej: Nota del examen
- Regresión lineal

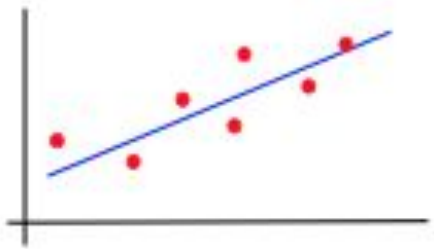
Clasificación y regresión (supervisado)

Clasificación





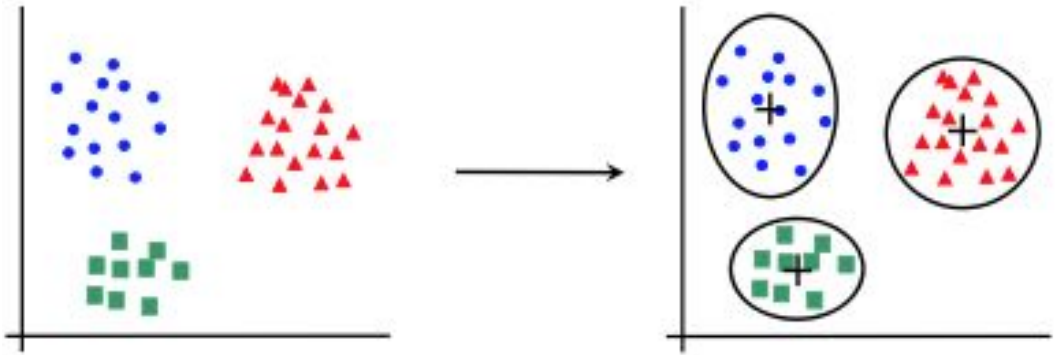
Regresión



Aprendizaje no supervisado (ya estudiado)

$$\{\mathbf{x}^{(i)}\} \propto p(x)$$

- aprender sobre *p*



Generalización

- No sólo buscamos que $f_{\boldsymbol{\omega}}(\mathbf{x}^{(i)}) \approx y^{(i)}$ (entrenamiento)
- Sino también $f_{\boldsymbol{\omega}}(\mathbf{x}^{(\mathbf{new})}) \approx y^{(\mathbf{new})}$ (test)

Paramétricos vs no paramétricos

Paramétricos: el modelo tiene un conjunto limitado de parámetros

- Regresión lineal
- Regresión logística
- Naïve Bayes
- Redes neuronales

- Eficientes: sencillos de entrenar
- Menos complejos

No paramétricos: la complejidad aumenta con el número de muestras

- Vecinos más próximos K-NN
- Kernel SVM
- Árboles de decisión

- Más flexibles
- Computacionalmente costosos

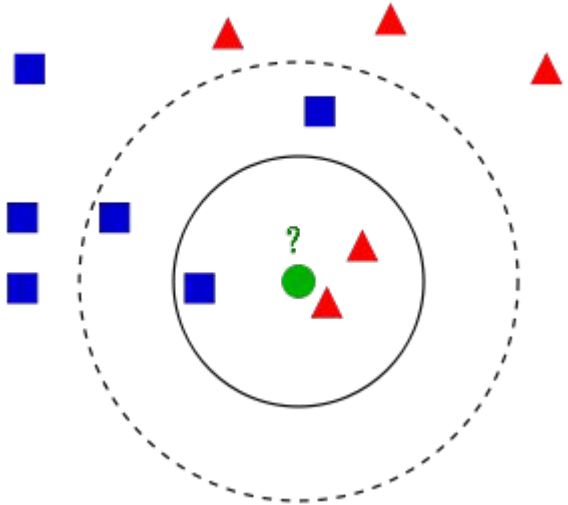
Índice

1. Introducción
2. Tipos de machine learning
3. **Vecinos más próximos**
    a. Evaluación y selección del modelo
4. ¿Cómo elegir el algoritmo adecuado?
5. Ciclo de vida de un proyecto en ML
6. Otras consideraciones

Vecinos más próximos (K-NN)

- Del inglés, *K-Nearest Neighbors*
- Puede utilizarse en **clasificación** y en regresión (más adelante)

- Si k=3: Rojo
- Si k=5: Azul



Matemáticamente:

$$f(\mathbf{x}_0) = y_i$$
$$i = \arg\min_j \left(||\mathbf{x}_j - \mathbf{x}_0||_2\right)$$

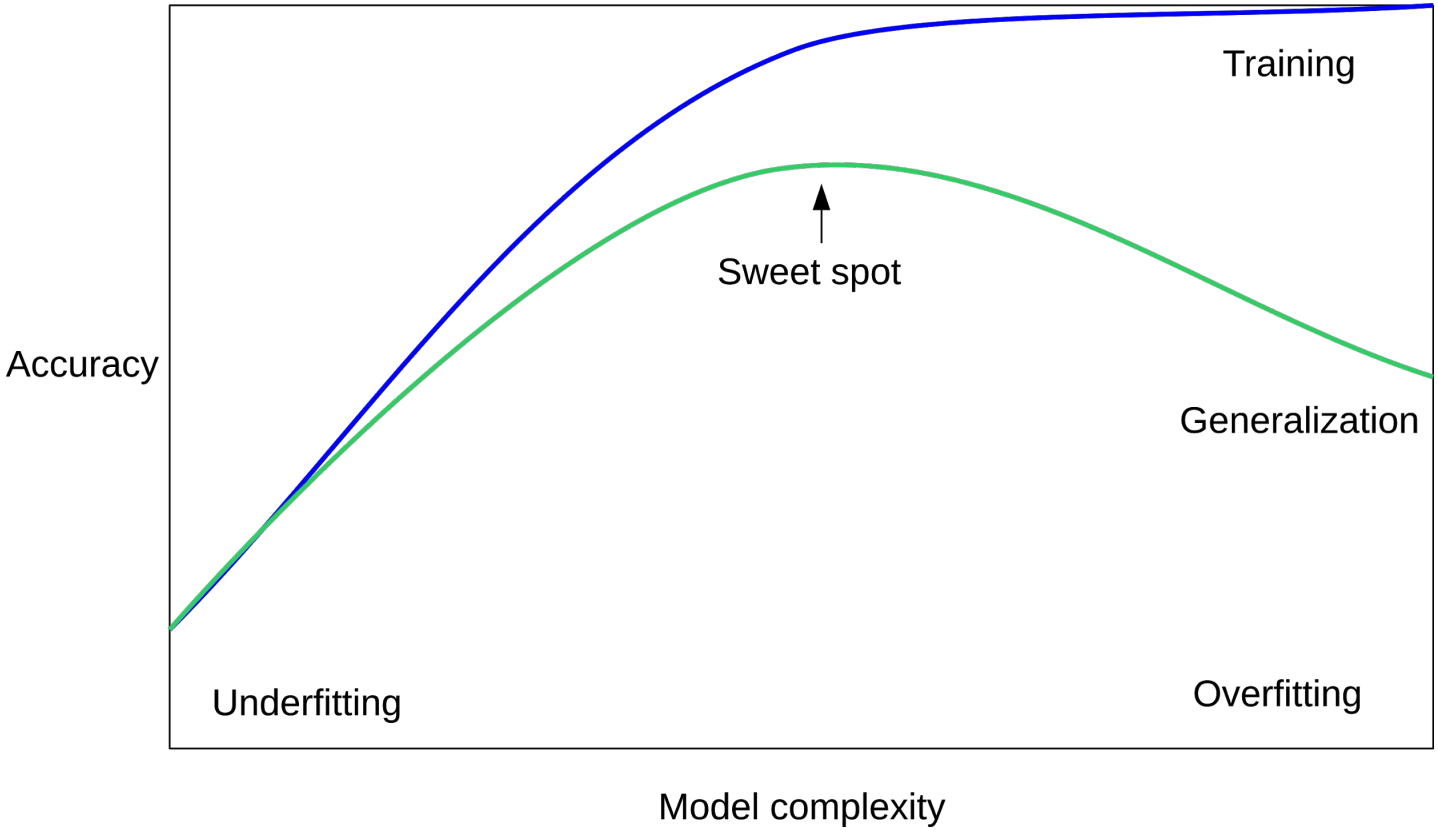
Fuente: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Hora de practicar

Índice

1. Introducción
2. Tipos de machine learning
3. Vecinos más próximos
    a. Evaluación y **selección del modelo**
4. ¿Cómo elegir el algoritmo adecuado?
5. Ciclo de vida de un proyecto en ML
6. Otras consideraciones

Train + test: sobreajuste

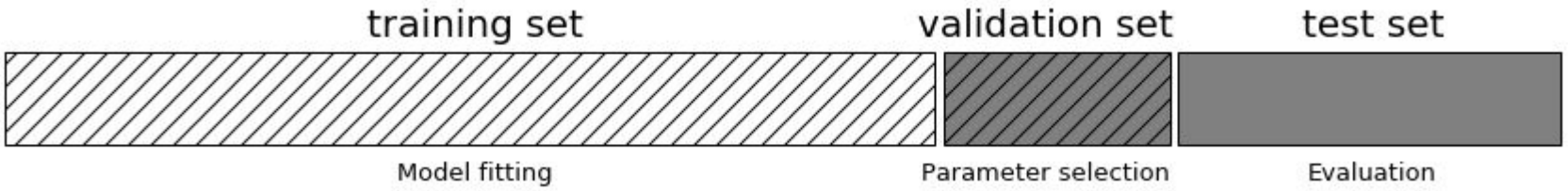


Limitaciones train + test

- Si las muestras de entrenamiento son escasas, el error en test puede ser muy variable, dependiendo de las muestras incluidas en el conjunto de entrenamiento y el conjunto de test.

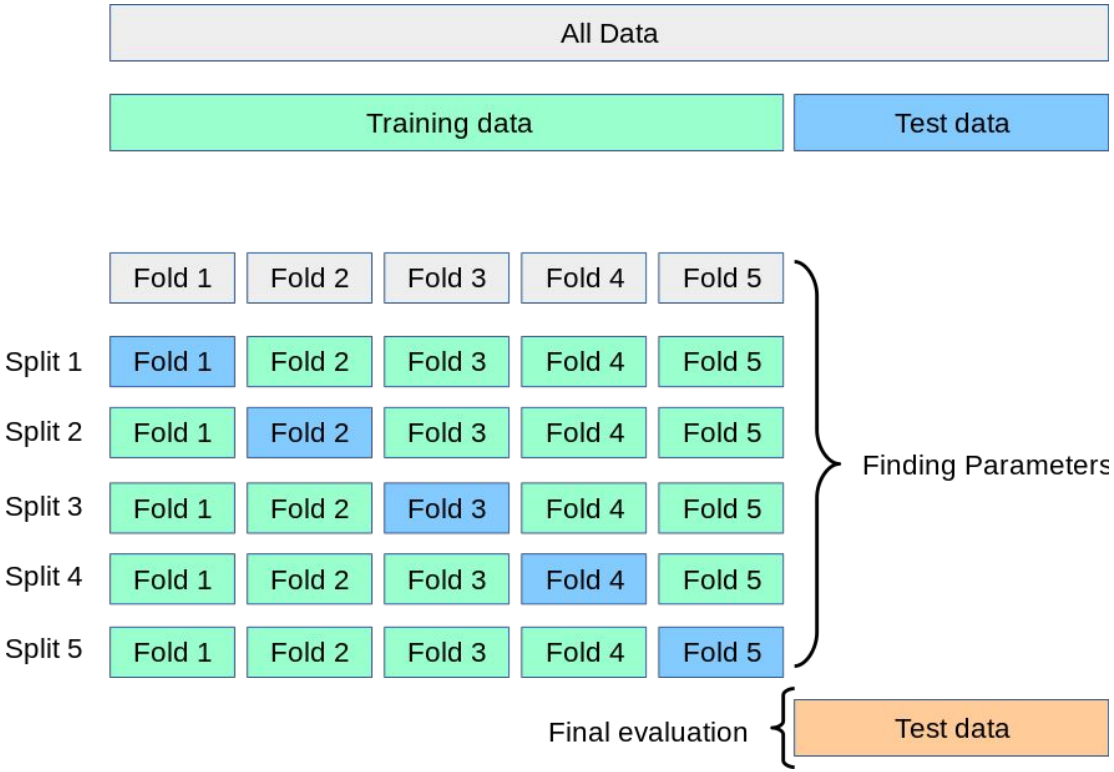
- No permite seleccionar los parámetros del modelo

Entrenamiento + validación + test

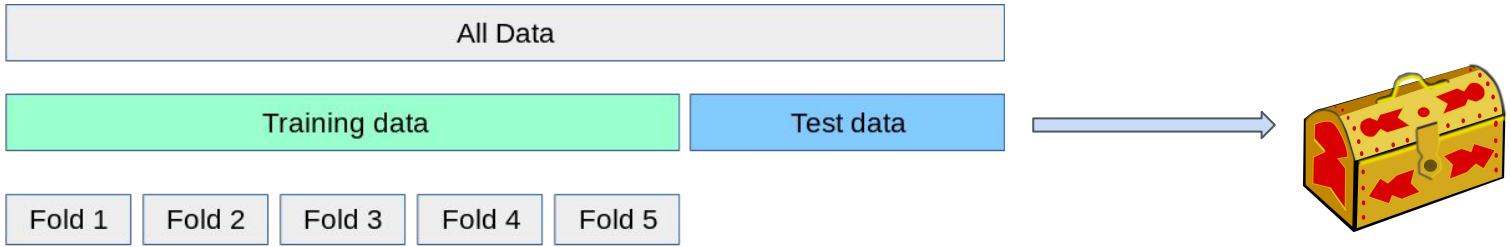


- Rápido y sencillo
- Mucha varianza (mismas limitaciones que caso anterior)

Validación cruzada: k-fold *cross-validation*



Validación cruzada: Paso 1



Validación cruzada: Paso 2

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5

for *n* = 1:Nvecinos

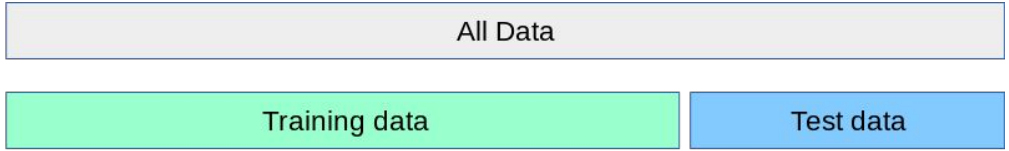
Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	$\text{Err}_{n,1}$
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	$\text{Err}_{n,2}$
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	$\text{Err}_{n,3}$
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	$\text{Err}_{n,4}$
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	$\text{Err}_{n,5}$

$$\text{Err}_n = \frac{1}{5}\sum_{i=1}^{5}\text{Err}_{n,i}$$

end

$$n_{opt} = \arg\min_n (\text{Err}_n)$$

Validación cruzada: Paso 3



$$n_{opt} = \arg\min_{n} \left(\mathrm{Err}_n\right)$$

$$\mathrm{Err}_{\mathrm{test}}$$

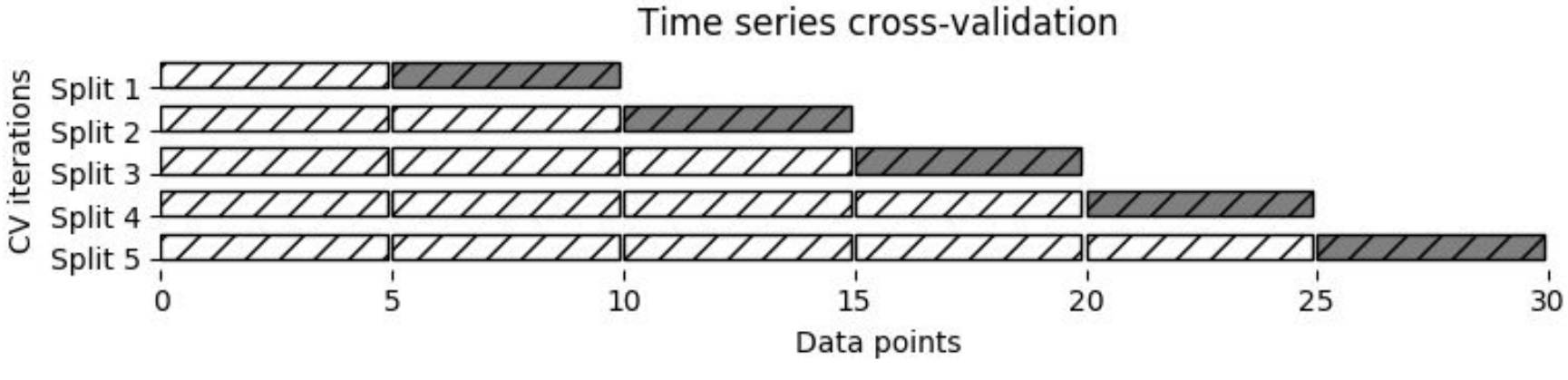
Consideraciones sobre k-fold *CV*

- Si K = N (número de muestras) se tiene ***leave-one out CV***
  - N-1 muestras para entrenar, y 1 muestra para medir prestaciones
  - El conjunto de entrenamiento es muy parecido para cada fold ⇒ la estimación del error de tiene poco sesgo, pero mucha varianza.
  - Es computacionalmente costoso

- En la práctica **K = 5, 10 proporciona buenos resultados**, buen compromiso entre sesgo y varianza

CV en series temporales

- No es un proceso i.i.d



Hora de practicar

Índice

1. Introducción
2. Tipos de machine learning
3. Vecinos más próximos
    a. Evaluación y selección del modelo
4. **¿Cómo elegir el algoritmo adecuado?**
5. Ciclo de vida de un proyecto en ML
6. ML en la vida real

¿Cómo elegir el algoritmo adecuado?

- No *free lunch*, no hay un algoritmo mejor que otro para todos los problemas

- "*All models are wrong, but some are useful*", George Box

Algunas consideraciones

- Compromiso sesgo-varianza
- Ruido y número de muestras de entrenamiento
- Complejidad de la solución
- Dimensionalidad del conjunto de entrada

Otros factores

- Heterogeneidad de los datos
  - Árboles vs algoritmos basados distancia
- Redundancia
  - Métodos lineales
- Interacciones y relaciones complejas

Hora de practicar

Índice

Principios del aprendizaje

- Navaja de Occam: el modelo más simple es el más plausible

- Sesgo en la población: el aprendizaje también estará sesgado

- Manipulación en el conjunto de test
  - Normalización de variables
  - Selección de características

Índice

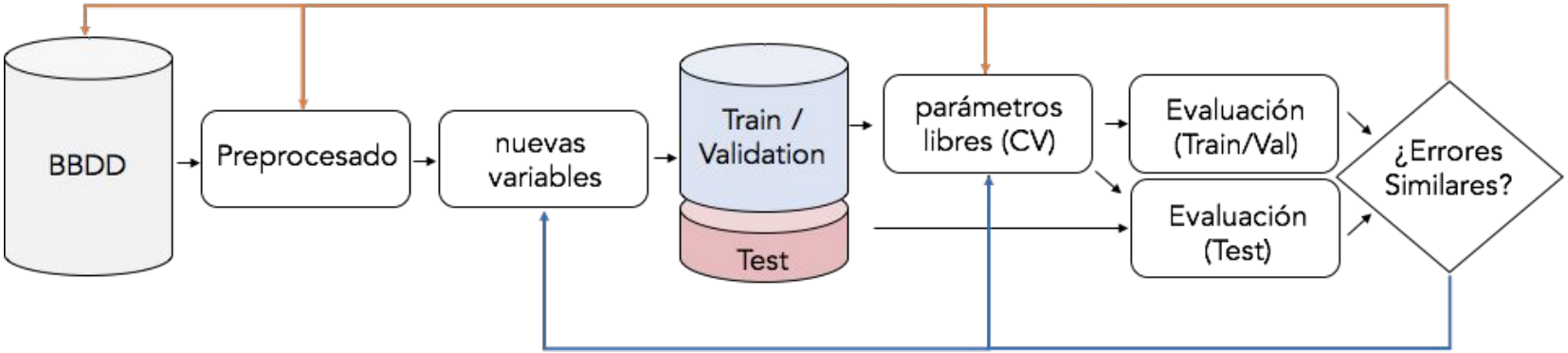
1. Introducción
2. Tipos de machine learning
3. Vecinos más próximos
    a. Evaluación y selección del modelo
4. ¿Cómo elegir el algoritmo adecuado?
5. Principios del aprendizaje
6. **Ciclo de vida de un proyecto en ML**
7. ML en la vida real

ML pipeline: general

ML pipeline: específico

Errores muy distintos (overfitting):

1. Conseguir más muestras de entrenamiento
2. Reducir el número de variables
3. Aumentar el valor del parámetro de regularización



Errores similares, pero de valor elevado:

1. Añadir nuevas variables
2. Añadir variables polinómicas y/ointeracciones
3. Disminuir el valor del parámetro de regularización

Índice

1. Introducción
2. Tipos de machine learning
3. Vecinos más próximos
    a. Evaluación y selección del modelo
4. ¿Cómo elegir el algoritmo adecuado?
5. Principios del aprendizaje
6. Ciclo de vida de un proyecto en ML
7. **ML en la vida real**

Principios básicos

- Definición del problema: elegir la tarea de ML adecuada
  - Probabilidad de que un cliente deje de usar la aplicación: ¿regresión, clasificación, clustering?
- Recopila datos, análisis exploratorio, y después (si es necesario), aplica ML (no comenzar con *deep learning*)
- Mide el impacto:
  - ¿De verdad necesitas un algoritmo de ML? ¿y qué beneficios vas a obtener? ¿y cómo mides esos beneficios?
- Explicar los resultados
  - Interpretabilidad y comunicación
  - Sistemas de recomendación mejoran si se dicen causas de recomendación

Referencias

- An Introduction to Statistical Learning.
  - Capítulos 2, 5.

- Machine Learning a Probabilistic Perspective.
  - Capítulo 1