keep coding

Machine Learning y Ciberseguridad

Fundamentos de Machine Learning

Índice

¿Qué es machine learning?

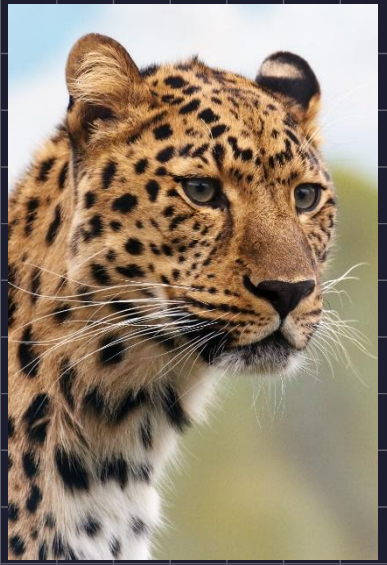
La ciencia de

- “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 si dispones de datos para construir una solución empírica

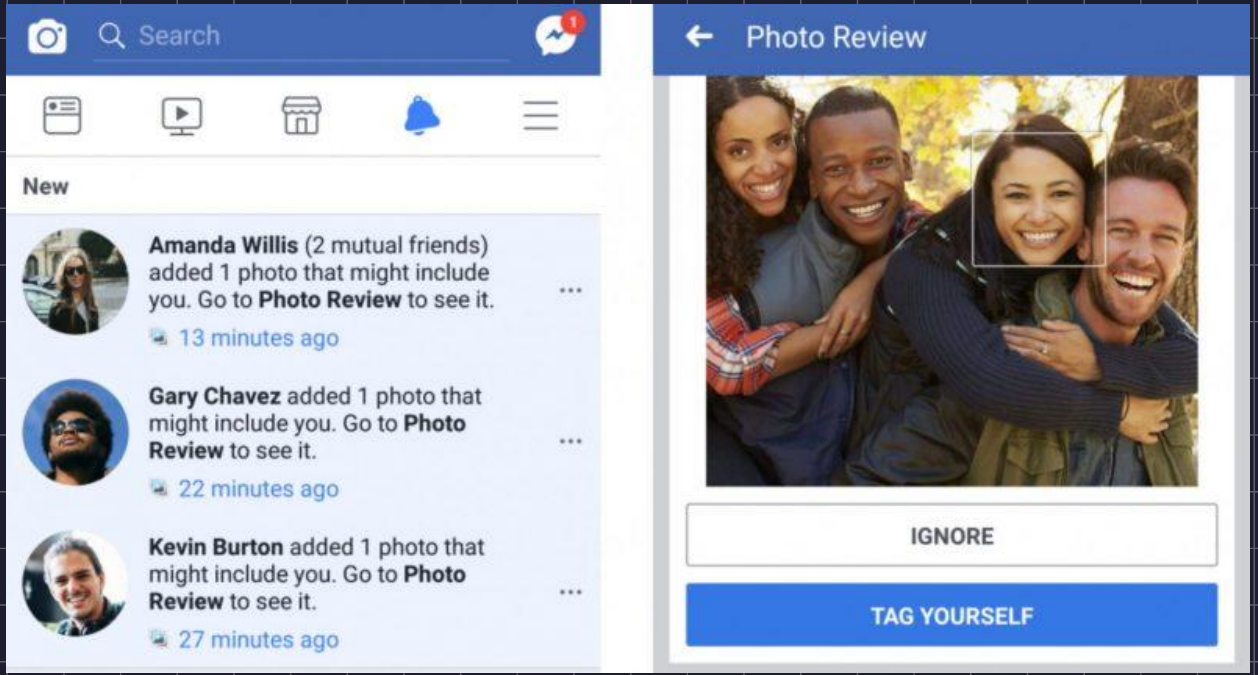
keep coding

¿Qué es machine learning?

¿Qué es un gato?



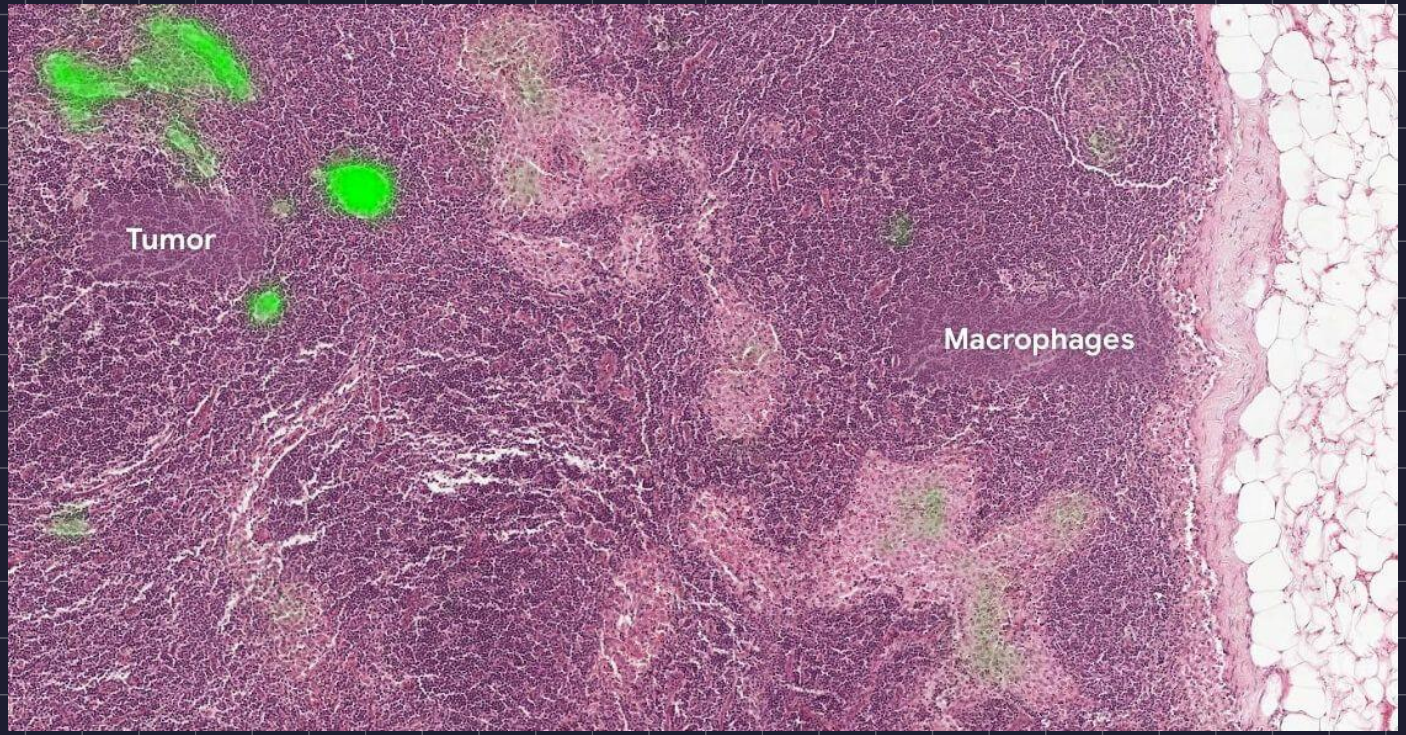
¿Qué es machine learning?



¿Qué es machine learning?

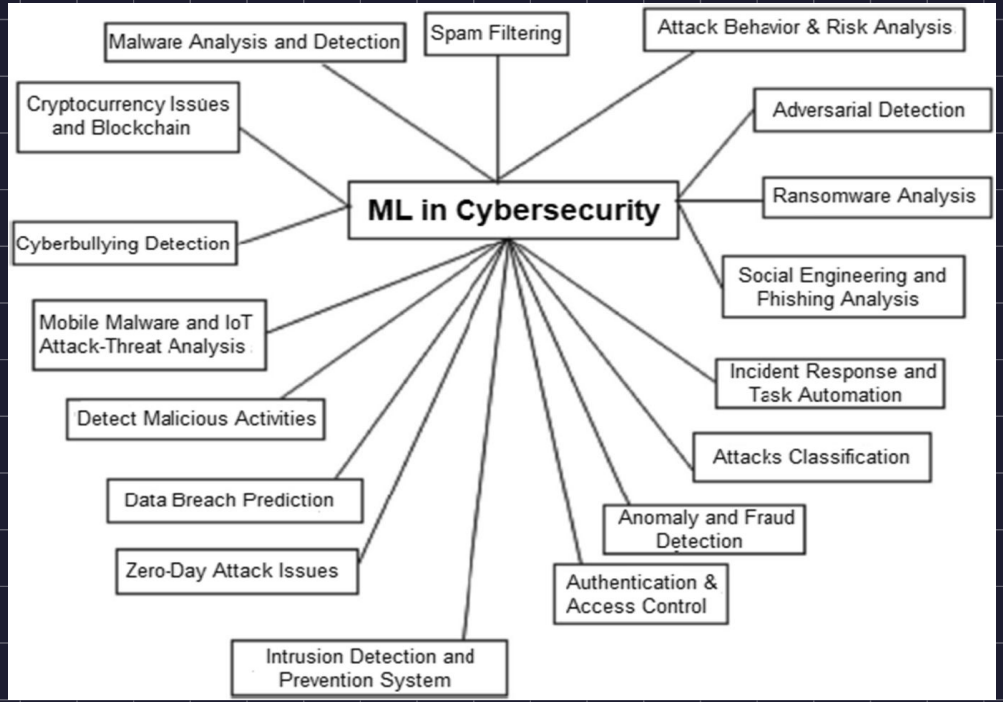
keep coding

¿Qué es machine learning?



keep coding

¿Qué es machine learning?



https://link.springer.com/article/10.1007/s40745-022-00444-2

¿Y qué NO es?
Diferencias con la IA

- La inteligencia artificial es la “ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes” – McCarthy, 1956

- La definición es difusa: “inteligencia” llevada a cabo por máquinas.
  - Técnicamente, la percepción del entorno y consecución de objetivos se considera inteligencia.
  - La definición más aceptada socialmente incluye funciones cognitivas: percepción, razonamiento, resolución.
  - ¿Pero qué es realmente la inteligencia?

keep coding

¿Y qué NO es?
Diferencias con la IA

- IA estrecha o narrow: Resuelve una tarea de forma igual o superior a un humano. DeepBlue (sin ML!), AlphaGo (RL). No va más allá de esa tarea; cualquier otra actividad escapa a su comprensión. AlphaGo es capaz de vencer a los grandes maestros del Go, pero no puede pedir una pizza. De hecho, ni siquiera sabe que está jugando al Go.

- IA general o AGI: Inteligencia a nivel humano. Según dicen, estamos cerca de alcanzarla; aunque hace dos décadas también decían que lo estábamos (spoiler: no lo estábamos)

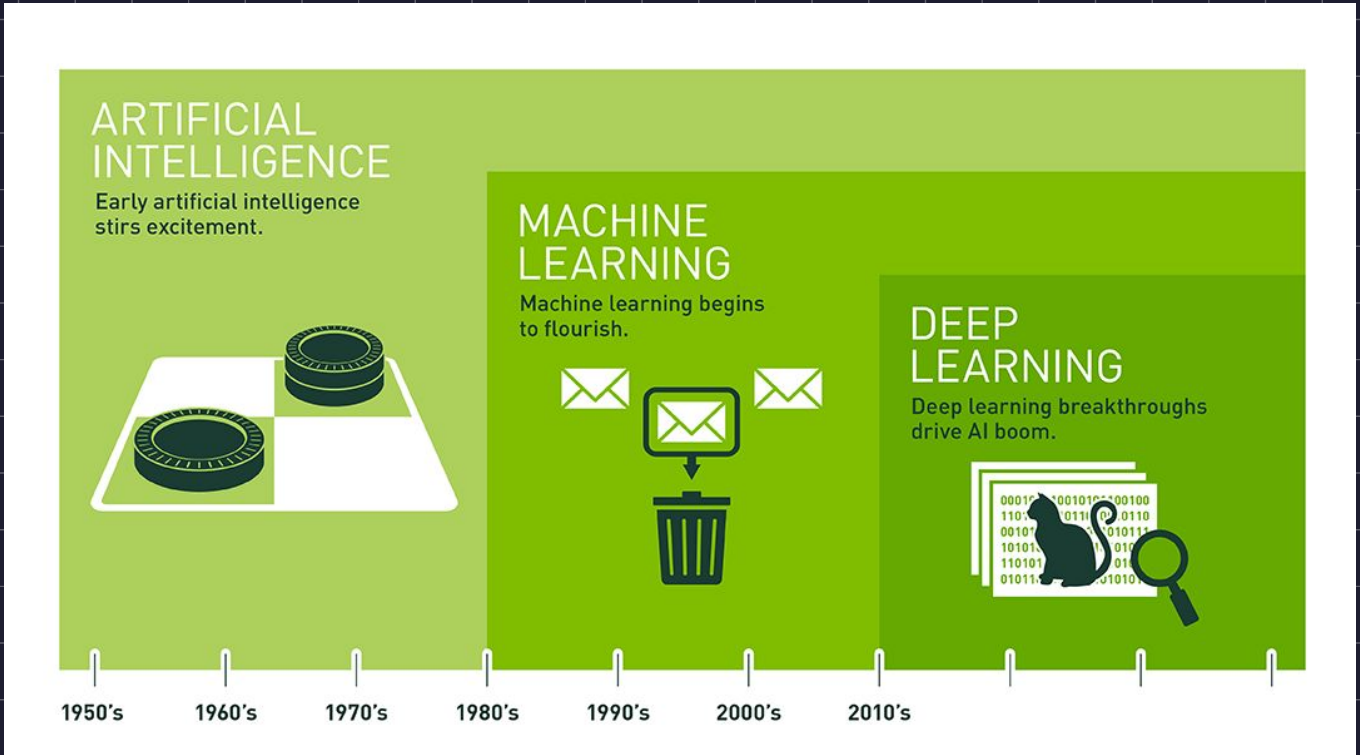
- Super inteligencia o ASI: Superior a los humanos en cualquier ámbito, incluyendo creatividad artística y habilidades sociales.

keep coding

Diferencias con Deep Learning

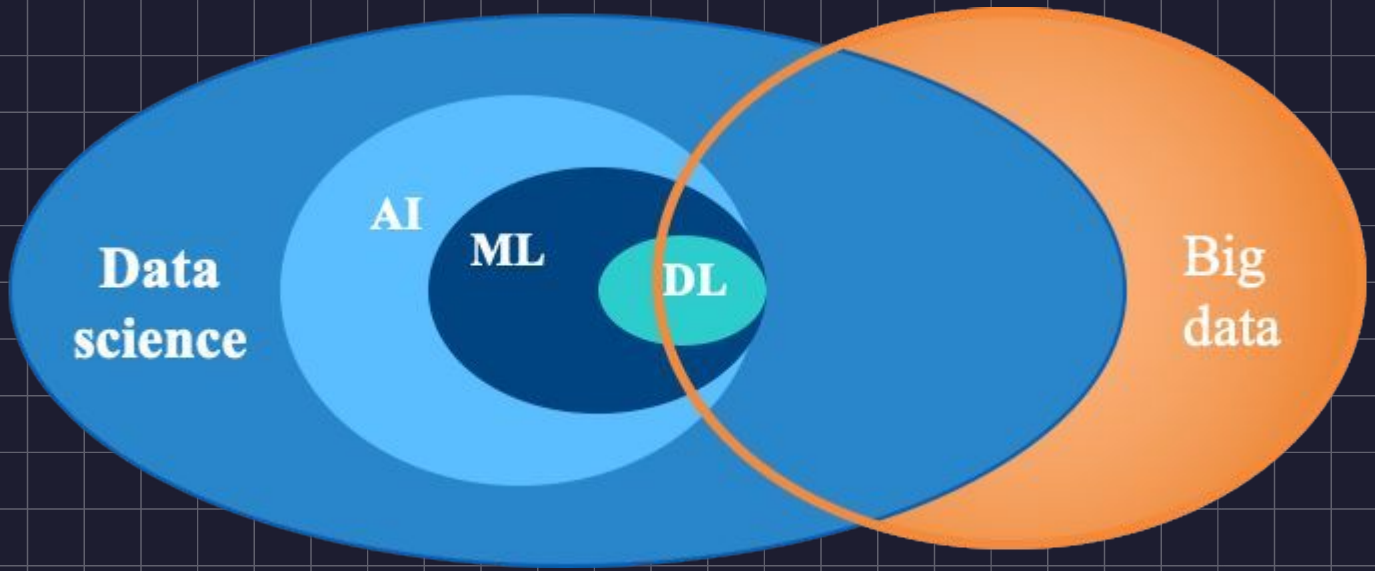
- Redes neuronales (algoritmo de machine learning)
- Arquitecturas complejas (profundas)
- Teorizadas en los años 50, recuperadas en 2010 gracias a GPUs y datos masivos (digitalización)
- Grandes resultados en datos no estructurados
  - Imagen médica
  - Gaming
  - Chatbots

AI, ML y DL


ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.
MACHINE LEARNING
Machine learning begins to flourish.
DEEP LEARNING
Deep learning breakthroughs drive AI boom.
1950's
1960's
1970's
1980's
1990's
2000's
2010's

keep coding

Relación entre ML y ciencia de datos



Estado actual

- Las empresas más grandes llevan algunos años con estas tecnologías implantadas; se van extendiendo paulatinamente.
- El impacto es real, pero hay humo. Mucho humo. Por todas partes. Hoy en día el uso de agentes de ML es masivo.

TECH \ ARTIFICIAL INTELLIGENCE

Forty percent of 'AI startups' in Europe don't actually use AI, claims report

Companies want to take advantage of the AI hype

By James Vincent | Mar 5, 2019, 8:14am EST

Fuente: https://www.theverge.com/2019/3/5/18251326/ai-startups-europe-fake-40-percent-mmc-report

¿Y en lo laboral?

	Business-Oriented	Engineering-Oriented
Emerging	• Data Analyst • Data Scientist • Data/ML Product Manager	• Data Engineer • ML Researcher/Scientist • ML/DL/AI Engineer
Traditional	• Business Analyst (Various Functions) • BI Analyst	• BI Engineer/Developer

DATAKADEMY

Fuente: https://hackernoon.com/navigating-the-data-science-career-landscape-db746a61ac62

Machine Learning en resumen

Un modelo de ML aprende de manera automática patrones en los datos. Una vez que ha aprendido, podemos utilizarlo para hacer predicciones sobre datos nuevos que nunca ha visto. Para ello:

- Necesitamos recopilar datos para entrenar el modelo
- Necesitamos seleccionar un modelo de ML especifico
- Entrenar el modelo con los datos
- Una vez entrenado podemos hacer predicciones sobre datos nuevos.

Esto es diferente a la programación convencional, en cuanto a que no tenemos por qué programar qué aspecto y forma tienen los patrones de los datos, el modelo lo hace por nosotros.
[Analogía profesor con la clase]

Ejemplo en vivo
https://teachablemachine.withgoogle.com/

keep coding

Elementos Esenciales

Variable Aleatoria

- Variable “Normal”: La de toda la vida

  X=2*4 -> X=8
  X=Y+2 -> si Y=1, X=3
  si Y=4, X=6
  X=? -> X=valor definido

- Variable Aleatoria: Variable usada en estadística y probabilidad

  X~Normal(0,1)
  X=0 o X=0.5 o X=-0.2231
  X=? -> X=valor indefinido

- Decimos que una v.a. sigue una **distribución** de valores y cada valor tiene una probabilidad de ocurrencia.

Variable Aleatoria

- Decimos que una v.a. (variable aleatoria)sigue una distribución de valores y cada valor tiene una probabilidad de ocurrencia.
  e.g: Lanzar una moneda al aire.

  X = resultado de lanzar una moneda

  Valores posibles -> X=0(cruz) o X=1(cara)

  Probabilidades -> P(X=0)=0.5 P(X=1)=0.5

DataFrame

Tabla de datos donde guardaremos las observaciones de nuestros experimentos. Es muy útil por que permite transformar y sacar información de los datos de manera sencilla y fácil.

lanzamiento	resultado
0	cara
1	cruz
2	cruz
3	cara
4	cruz
5	cruz
6	cara
7	cruz

```
df["valor"] = df["resultado"].map({"cara":1,"cruz":0})
df
```

lanzamiento	resultado	valor
0	cara	1
1	cruz	0
2	cruz	0
3	cara	1
4	cruz	0
5	cruz	0
6	cara	1
7	cruz	0

```
df.describe()
```

	valor
count	8.000000
mean	0.375000
std	0.517549

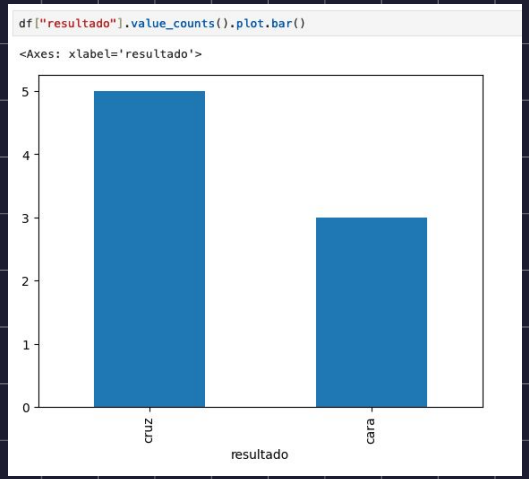
keep coding

DataFrame
Representación Visual

Además usar dataframes nos facilita mucho la vida el graficar los datos. Una tarea imprescindible.

Aquí podemos ver la representación visual en gráfico de barras del dataframe anterior.

Cada barra representa en número de veces que apareció cara o cruz.



keep coding

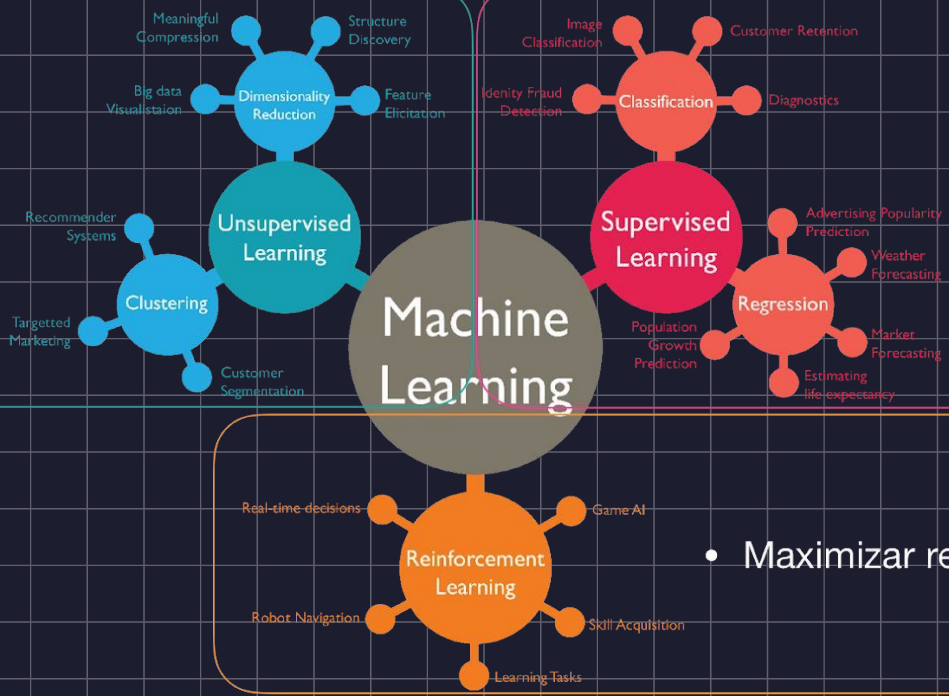
LET'S CODE

Índice

keep coding

Tipos de machine learning

- Agrupar
- Comparar
- Recomendar



- Clasificar
- Detectar
- Estimar
- Predecir

- Maximizar recompensa

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

- La variable objetivo *y* es discreta
- Ej: Apto / No apto
- Regresión logística

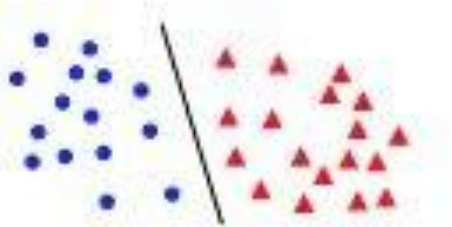
Regresión

- La variable objetivo *y* es continua
- Ej: Nota del examen
- Regresión lineal

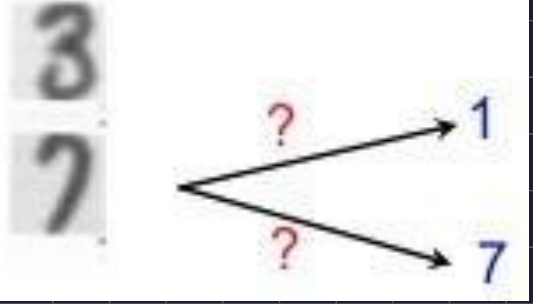
Clasificación y regresión (supervisado)

Clasificación

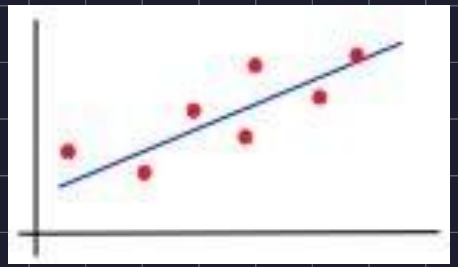
Binaria



Multiclase



Regresión



keep coding

LET'S CODE

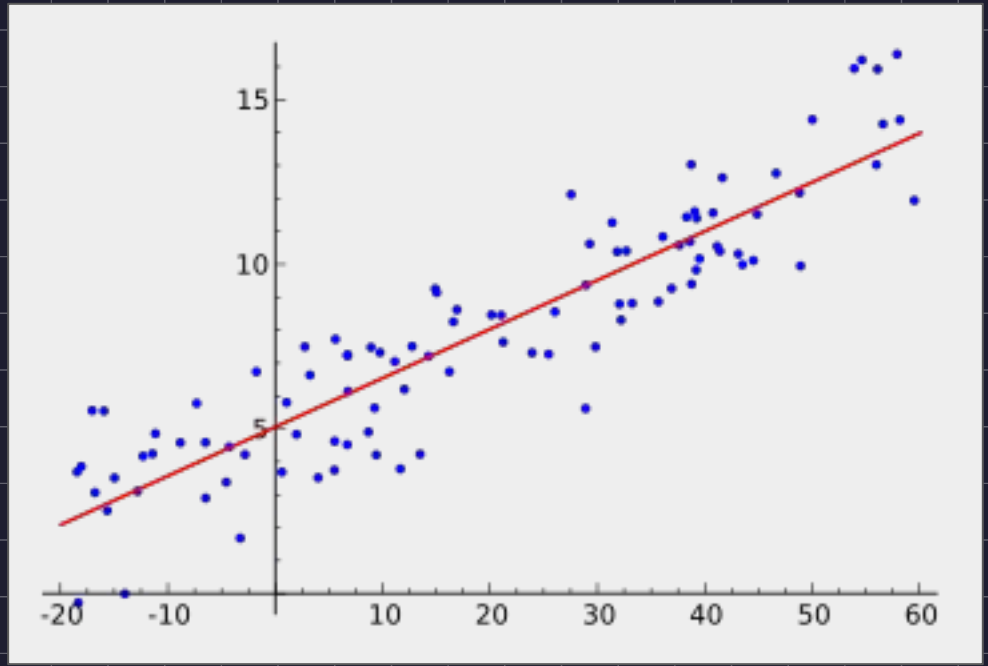
keep coding

Regresión Lineal

Nos será útil cuando los datos sigan una distribución lineal.

y=ax+b
a=pendiente, b=sesgo
e.g
a=0.2
b=5

https://www.desmos.com/calculator

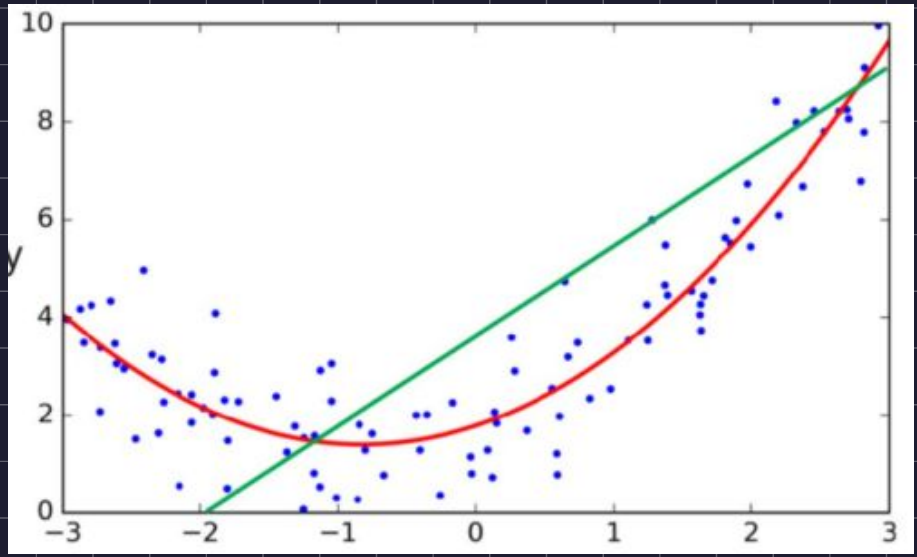


keep coding

Regresión Lineal grado 2

Nos será útil cuando los datos sigan una distribución curva simple

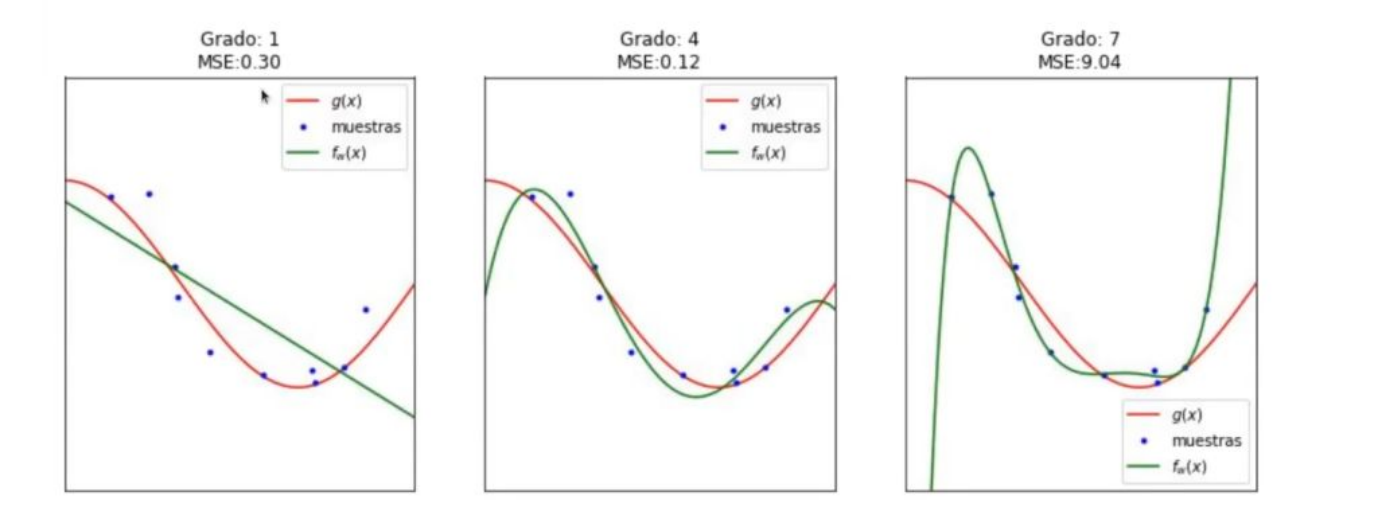
y=ax^2+bx+c



https://www.desmos.com/calculator

Regresión Lineal grado N

Nos será útil cuando los datos sigan una distribución curva compleja.
También llamada Regresión Polinomial
y=x^n+…+x^3+x^2+x+c



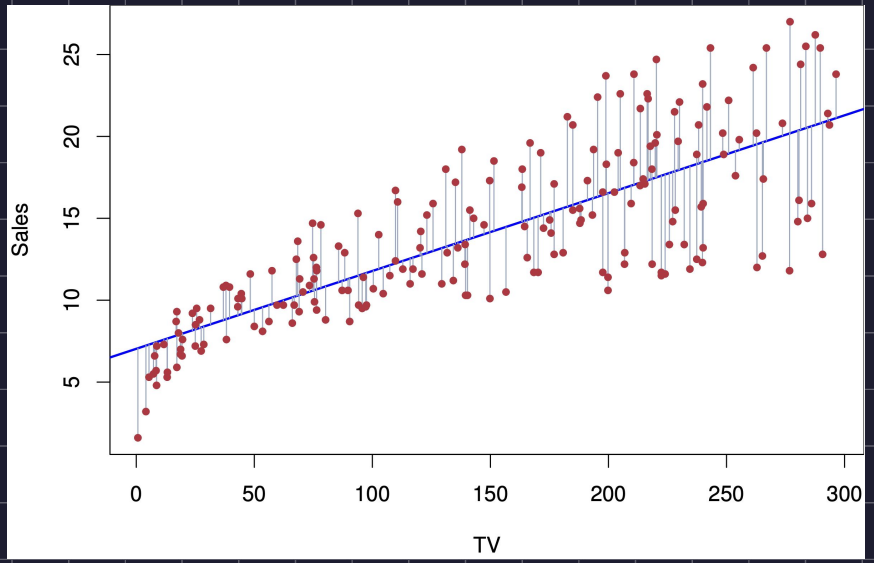
Error R2

Cómo podemos saber que modelo es mejor? -> El que menor error nos de.

El error es igual a la suma de todas las distancias en vertical de nuestra recta a los puntos de nuestros datos

$$y = g(x) + error$$

El error se suele medir con una métrica llamada R2, que nos da el valor de error entre 0 y 1. Siendo 1 = una regresión perfecta

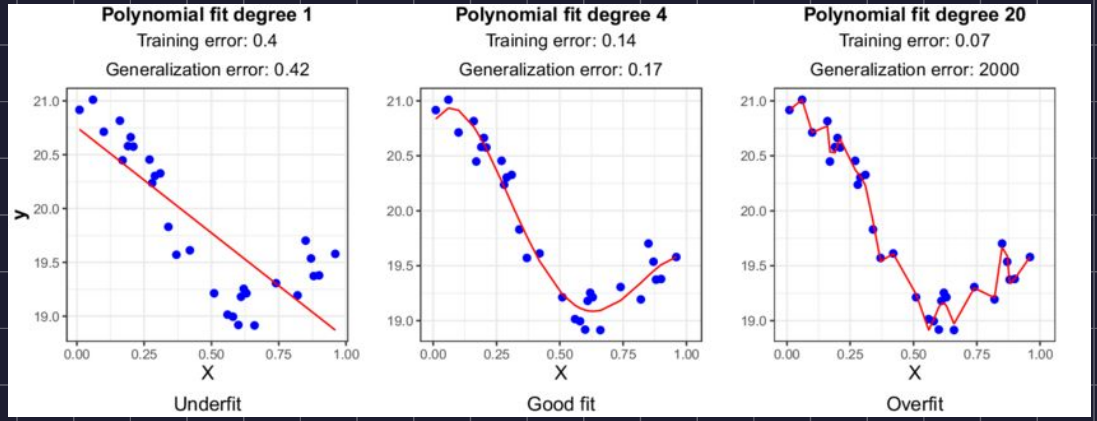


Error R2

Cómo podemos saber que modelo es mejor? -> El que menor error nos de.

El error es igual a la suma de todas las distancias en vertical de nuestra recta a los puntos de nuestros datos

$$y = g(x) + error$$



keep coding

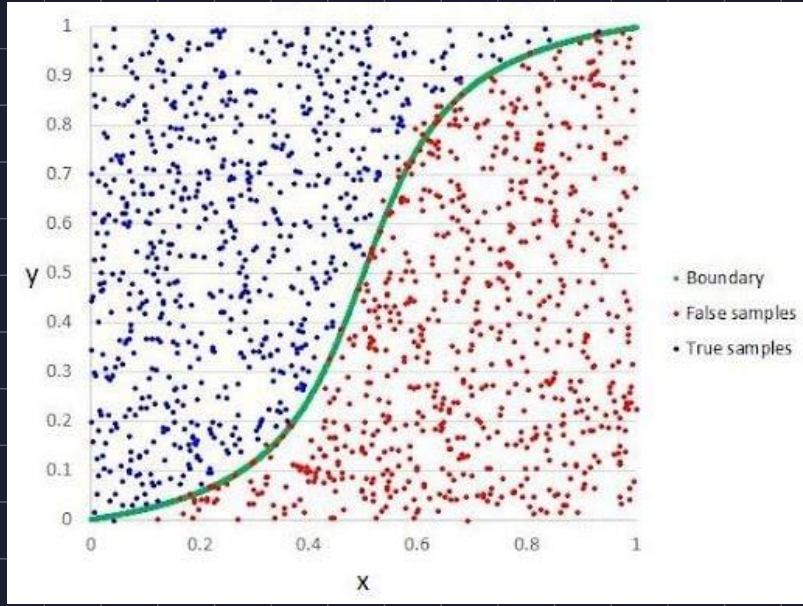
LET'S CODE

Regresión Logística

Nos será útil cuando los datos sigan una distribución lineal y **queramos clasificar entre dos categorías.**

Aunque tiene unas propiedades Interesantes, no lo veremos en este modulo ya que su utilidad en el mundo real es bastante reducido para la tarea que nos ocupa.

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression().fit(X, y)
```



Métricas y selección de modelo

Como hemos visto en el notebook, aumentar el grado del polinomio de nuestra regresión no siempre es la mejor idea.

¿Que pasa si nos llegan datos nuevos?
¿Se adaptaran igual de bien al modelo?

Como no podemos adivinar qué datos nuevos nos llegarán, tenemos que dividir los datos que ya tenemos en dos conjuntos

- Train -> Entrenaremos nuestro modelo con este conjunto
- Test -> No entrenaremos pero lo usaremos para medir la generalización del modelo

keep coding

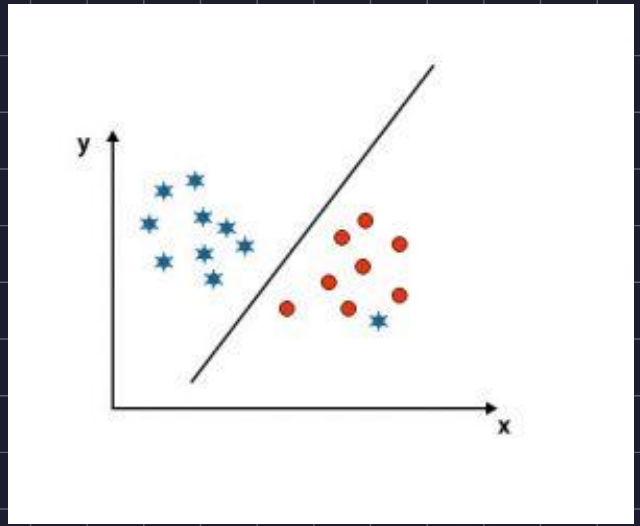
Generalización

No solo buscamos que el entrenamiento tenga buen resultado

$$f_\omega\left(x^{(i)}\right) \approx y^{(i)}$$

También que lo tenga el subconjunto de test

$$f_\omega\left(x^{(new)}\right) \approx y^{(new)}$$



keep coding

Métricas y selección de modelo

Una vez que ya tengamos nuestro conjunto de train y test, necesitaremos una métrica con la que medir la generalización.
Esta métrica varía dependiendo del tipo de variable a predecir, de momento usaremos dos.

Regresión
Root Mean Square Error (RMSE)
(Mientras mas bajo, mejor)

$$\sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Clasificación
Accuracy
(Mientras más alto, mejor)

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

keep coding

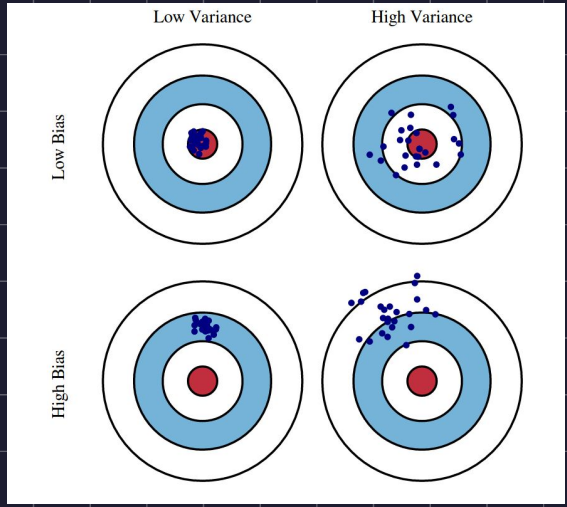
LET'S CODE

Sesgo VS Varianza

Llegados a este punto, hay que ser conscientes de la disyuntiva existente en ML:

- O asumimos principios (reales o no) sobre nuestros datos, reduciendo su error pero obteniendo resultados imprecisos con la realidad, **SESGADOS** (segunda fila)

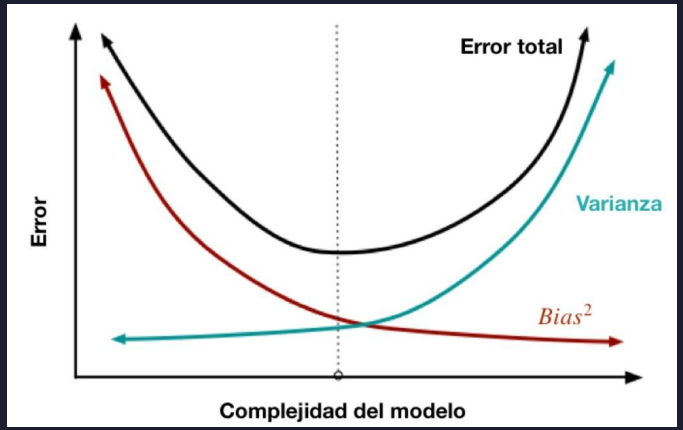
- O no asumimos nada, y obtenemos resultados con **ALTA VARIANZA** lo que nos hará obtener resultados menos precisos pero a la vez, “más cercanos” a la realidad (segunda columna)



Sesgo VS Varianza

No existe una verdad única en este sentido, pero hay una estrategia que podemos seguir. Elegir el equilibrio entre Sesgo y Varianza que nos de como resultado el menor error posible ya que la mayoría de las veces tendremos un objetivo concreto,

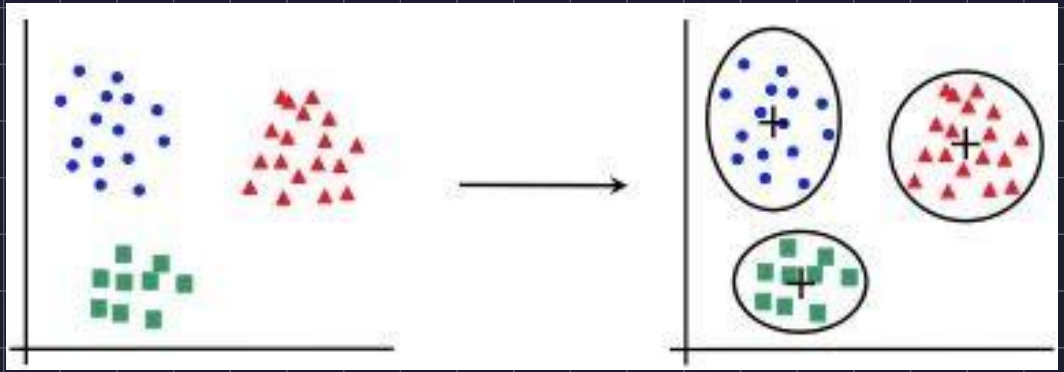
¿Cual es ese objetivo?



keep coding

Aprendizaje no supervisado

aprender sobre la distribución de los datos



Paramétricos vs no paramétricos

Paramétricos: el modelo tiene asunciones sobre cómo se generan los datos (normalmente, una distribución subyacente)

- Regresión lineal
- Regresión logística
- SVM with Kernel

- Permiten un análisis posterior
- Menos complejos, más robustos

No paramétricos: el modelo no tiene asunciones previas sobre los datos o estas son muy generales

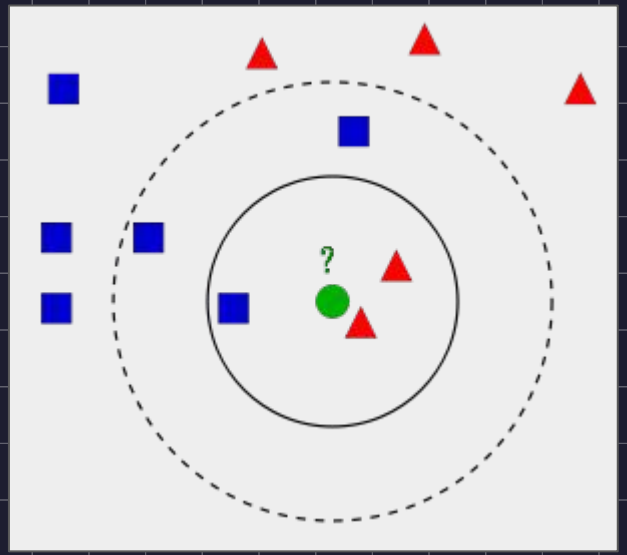
- Vecinos más próximos K-NN
- Árboles de decisión
- Naïve Bayes
- Redes neuronales

- La capacidad de aprendizaje está acotada por el poder de cómputo
- Son más complejos de entrenar e interpretar

Índice

Vecinos más próximos (K-NN)

- Del inglés, *K-Nearest Neighbors*
- Puede utilizarse en **clasificación** y en regresión



- ❑ Si k=3: Rojo
- ❑ Si k=5: Azul

Básicamente:

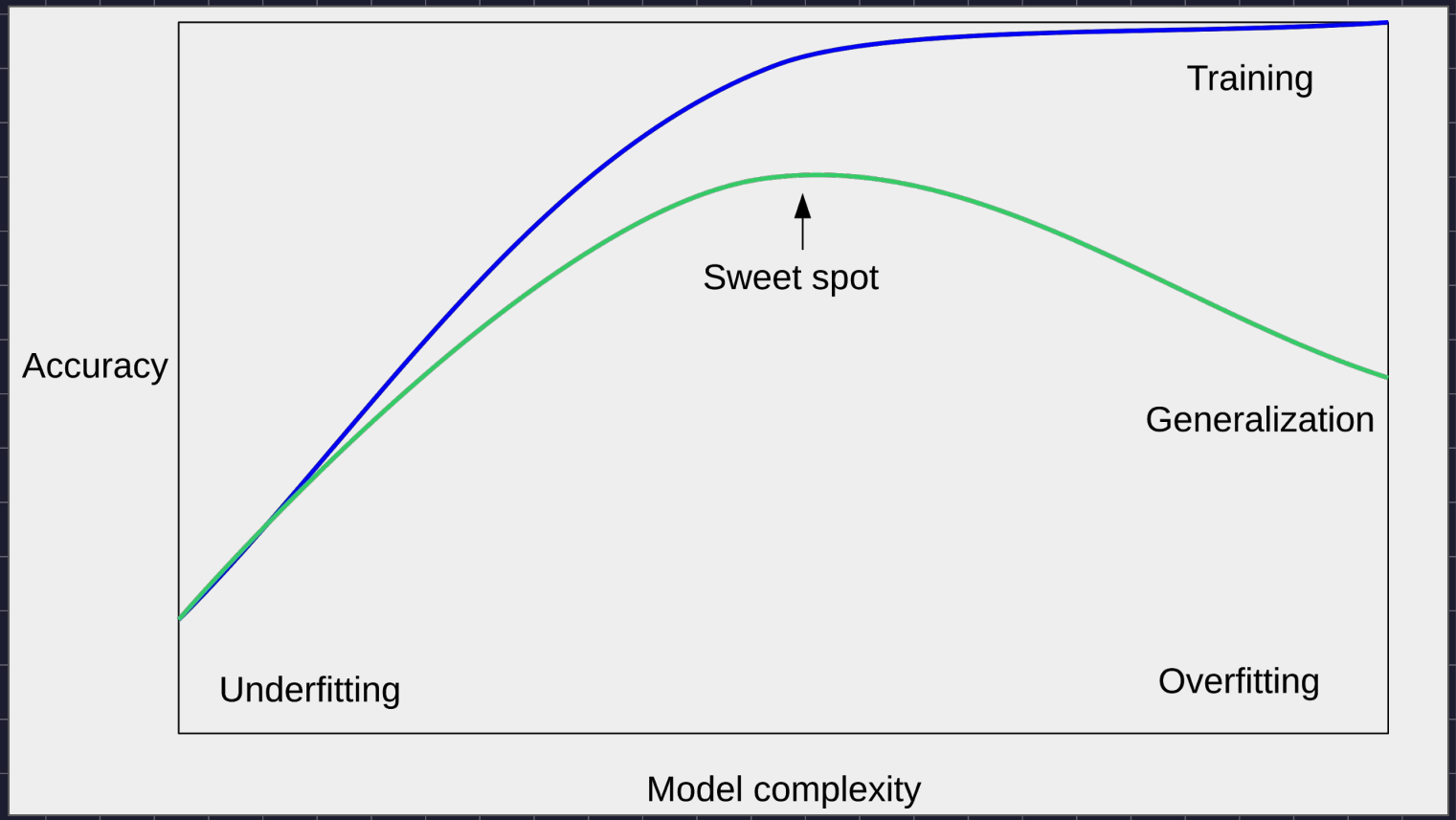
- En clasificación: voto de mayoría (moda)
- En regresión: media

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

keep coding

LET'S CODE

Train + test: overfitting



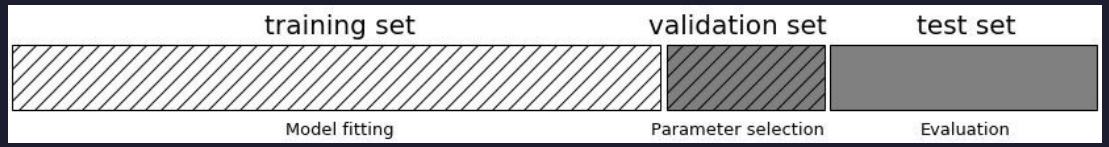
Limitaciones train + test

- 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.

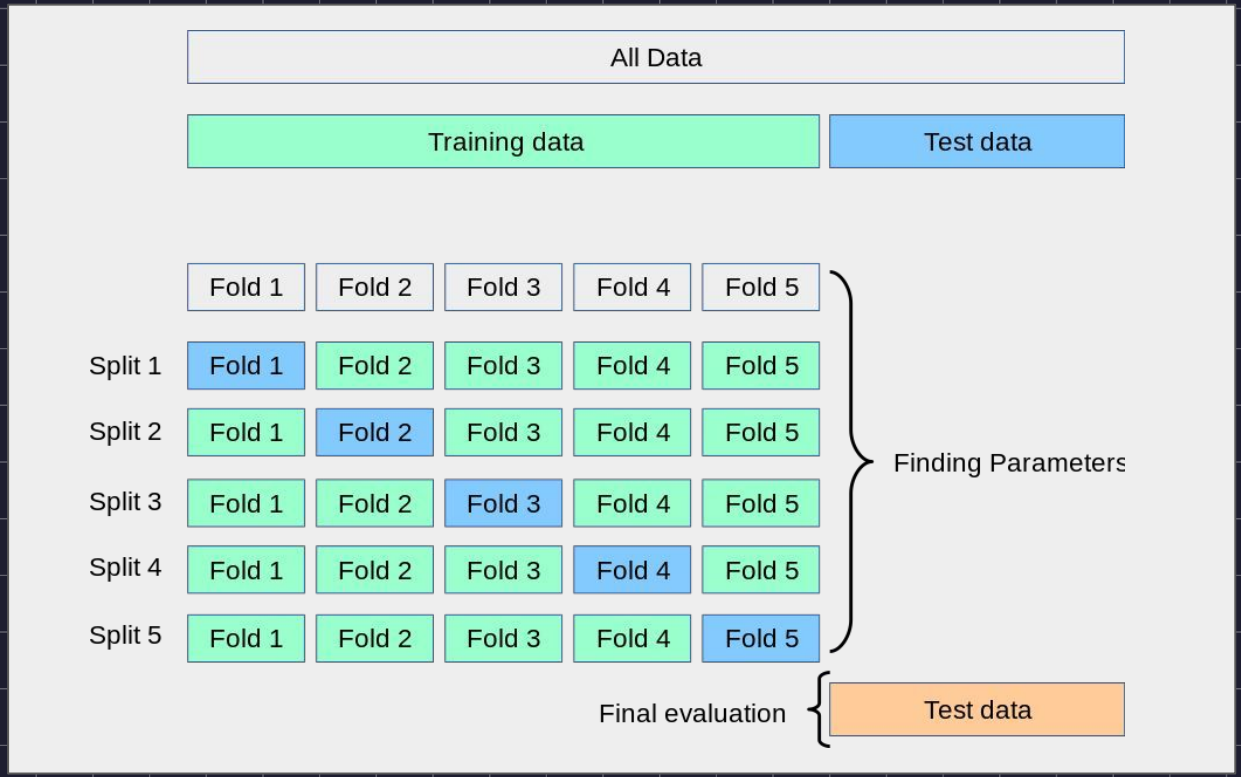
- No permite seleccionar los parámetros del modelo

Entrenamiento + validación + test

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



Validación cruzada: k-fold cross-validation



keep coding

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

keep coding

LET'S CODE

Índice

¿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
- Interacciones y relaciones complejas, no lineales
- Heterogeneidad de los datos
    - Árboles vs algoritmos basados en distancia

Índice

keep coding

Principios del aprendizaje

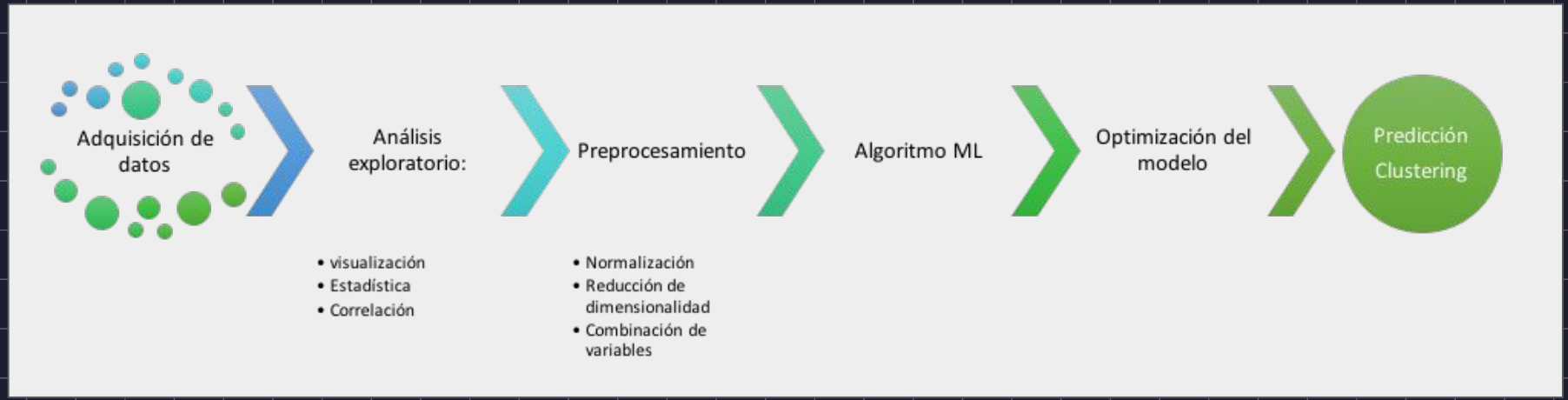
- Navaja de Occam: el modelo más simple es el más plausible con la realidad

- Sesgo en la población: el aprendizaje también estará sesgad

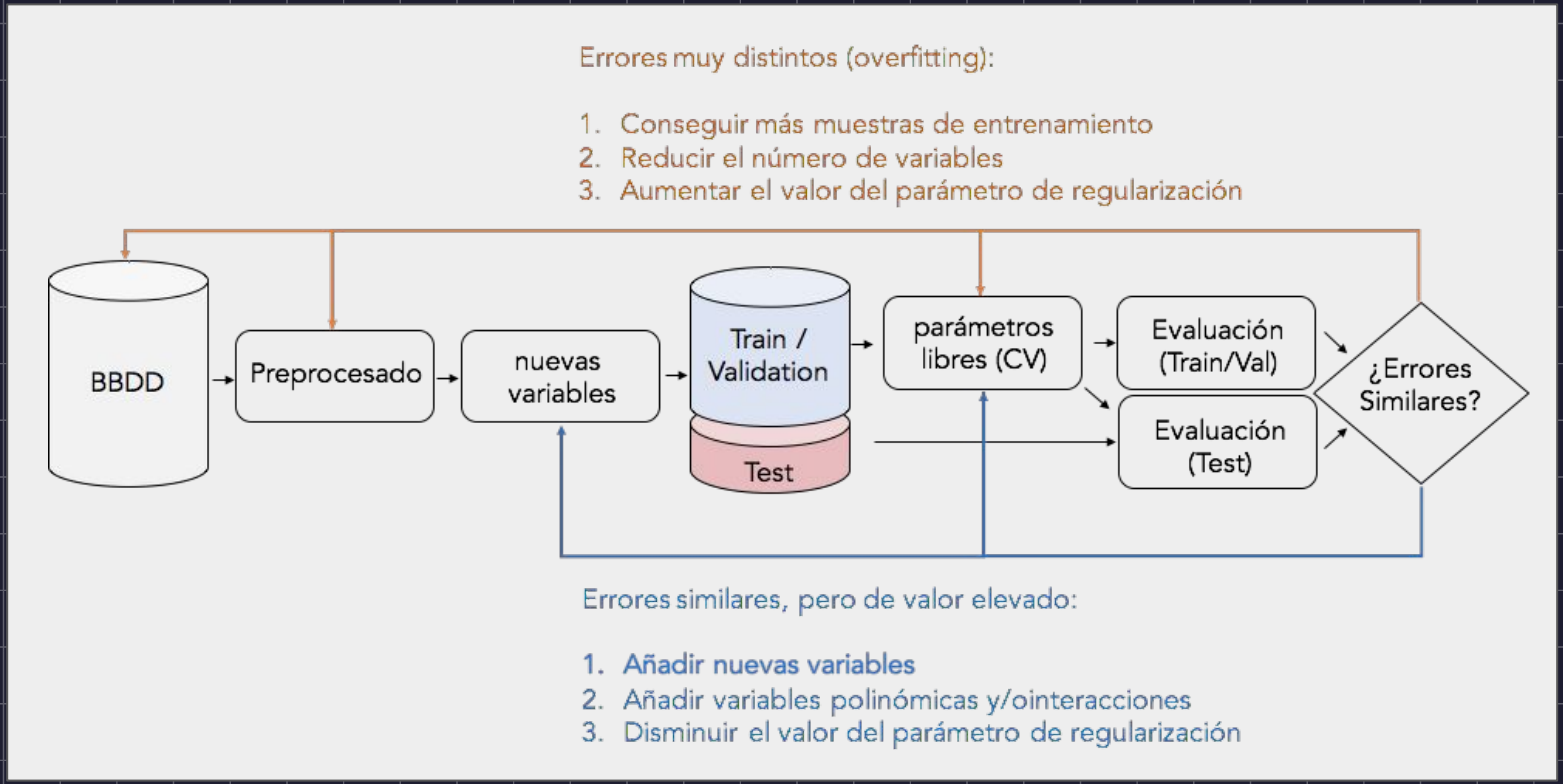
  - Manipulación en el conjunto de test (afecta el aprendizaje)
    - Normalización de variables
    - Selección de características

Índice

ML pipeline: general



ML pipeline: específico



Índice

ML en la vida real (cheatsheet)

- 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
- Mide el impacto:
  - ¿De verdad necesitas un algoritmo de ML? ¿y qué beneficios vas a obtener? ¿y cómo mides esos beneficios?
- Explica los resultados
  - Interpretabilidad y comunicación
  - Sistemas de recomendación mejoran si se dicen causas de recomendación

keep coding

LET'S CODE

keep coding

Referencias

- An Introduction to Statistical Learning.
  - Capítulos 2, 5.
- Machine Learning a Probabilistic Perspective.
  - Capítulo 1
- Hands On Machine Learning.
  - Capítulo 1

www.keepcoding.io cursos@keepcoding.io (+34) 916 33 1779