



Argentina  
programa  
4.0



Universidad  
Nacional  
de San Martín

# Módulo 2 - Ciencia de Datos

Semana 10

Repaso General



**Argentina  
programa  
4.0**

# PLAN DE ESTUDIOS

Introducción

## Programación en Python

Dictado del 13/02 al 19/04. **Lunes y  
Miércoles de 18 a 20 hs.**

[Ver mas](#)

Intermedio

## Ciencia de Datos

Dictado del 8/05 al 12/07. **Lunes y  
Miércoles de 18 a 20 hs.**

[Ver mas](#)

Especialización

## Aprendizaje Automático

Dictado del 7/08 al 16/10. **Lunes y  
Miércoles de 18 a 20 hs.**

[Ver mas](#)



Universidad  
Nacional  
de San Martín



Escuela de  
Ciencia y Tecnología  
ECyT\_UNSAM



Secretaría de Economía  
del Conocimiento

# ¿Qué vimos en este módulo?

- Conceptos básicos de matemática y probabilidad ¿Qué es un modelo?
- Análisis exploratório de datos
- Visualización de grandes conjuntos de datos
- Reduc. Dimensionalidad (PCA, t-SNE)
- Preprocesado de datos
- Agrupación/*Clustering*

## Regresión

- Modelos Predictivos/Explicativos
- Modelo de Regresión Lineal
- Modelo Polinomial: Feature Engineering
- Metricas
- Over/underfitting
- Regularización (L1 y L2)
- Cross Validation
- GridSearch (ejemplo: ploteos M vs Alpha)

## Clasificación:

- Función de pérdida
- Perceptrón
- Regresión logística
- Métricas de clasificación (matriz de confusión, P, R, etc.)
- Umbral (nuevo hiperparametro), ROC, P-R curve
- Práctica de clasificación
- Loss matrix
- Class Imbalance (usar F1)
- Class weights
- Clasificación multi-clase

# Contenidos del módulo

## Herramientas

- Fundamentos de análisis de datos.
  - Análisis exploratorio de datos
  - Limpieza de datos y preprocesamiento

- Modelo lineales de regresión
  - regresión lineal
  - regresión polinomial

- Modelos lineales para clasificación.
  - perceptron
  - Regresión Logística

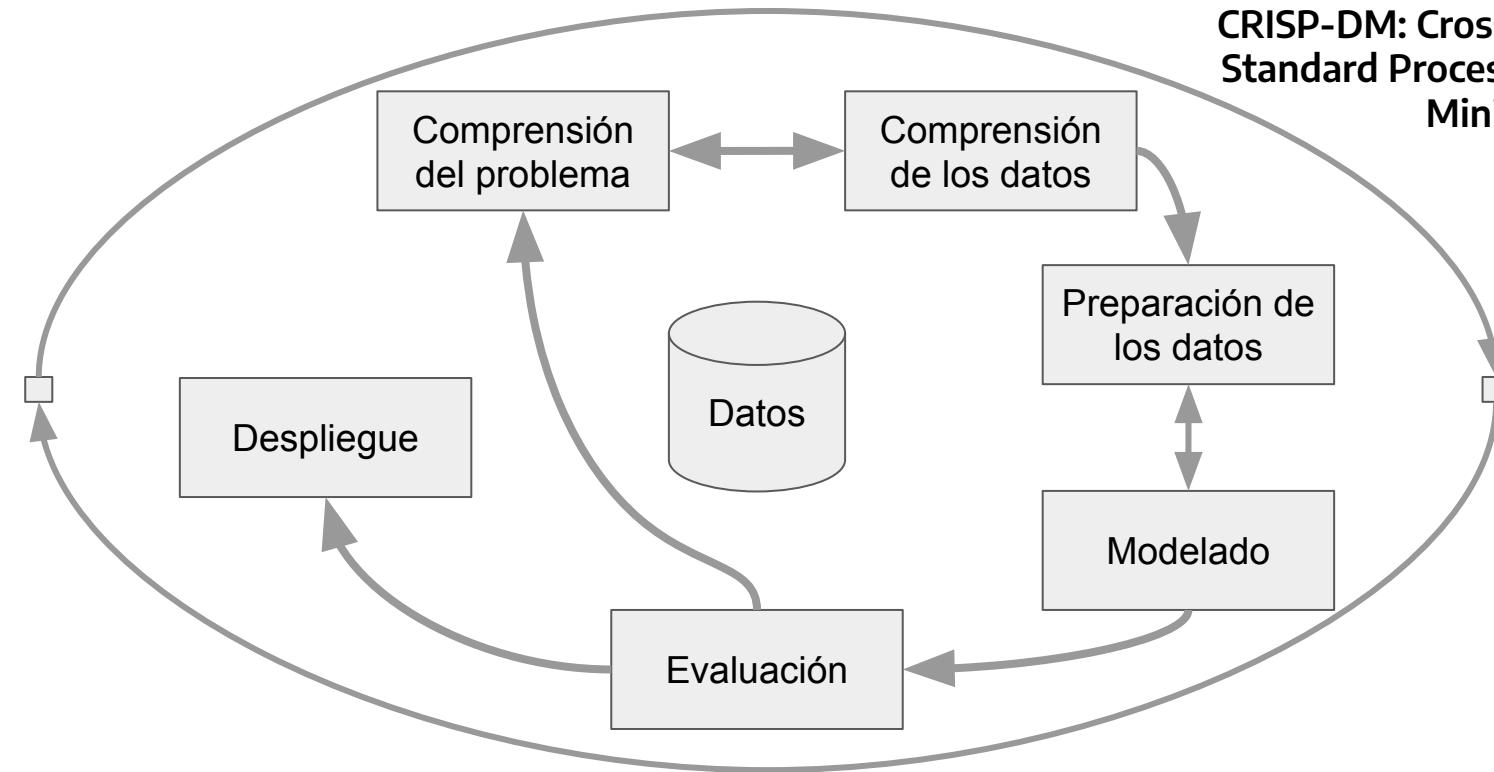
- Aplicaciones (a lo largo del curso)

## Conceptos

- Pensamiento orientado a datos
- Workflow de ciencia de datos
- Under / over-fitting
- Train / test sets.
- Conjunto de validación y validación cruzada.
- Fronteras y regiones de decisión
- Toma de decisiones
- Clasificación probabilística

# El proceso de ciencia de datos

CRISP-DM: Cross-Industry Standard Process for Data Mining (2006)



# El proceso de ciencia de datos

CRISP-DM: Cross-Industry Standard Process for Data Mining (2006)

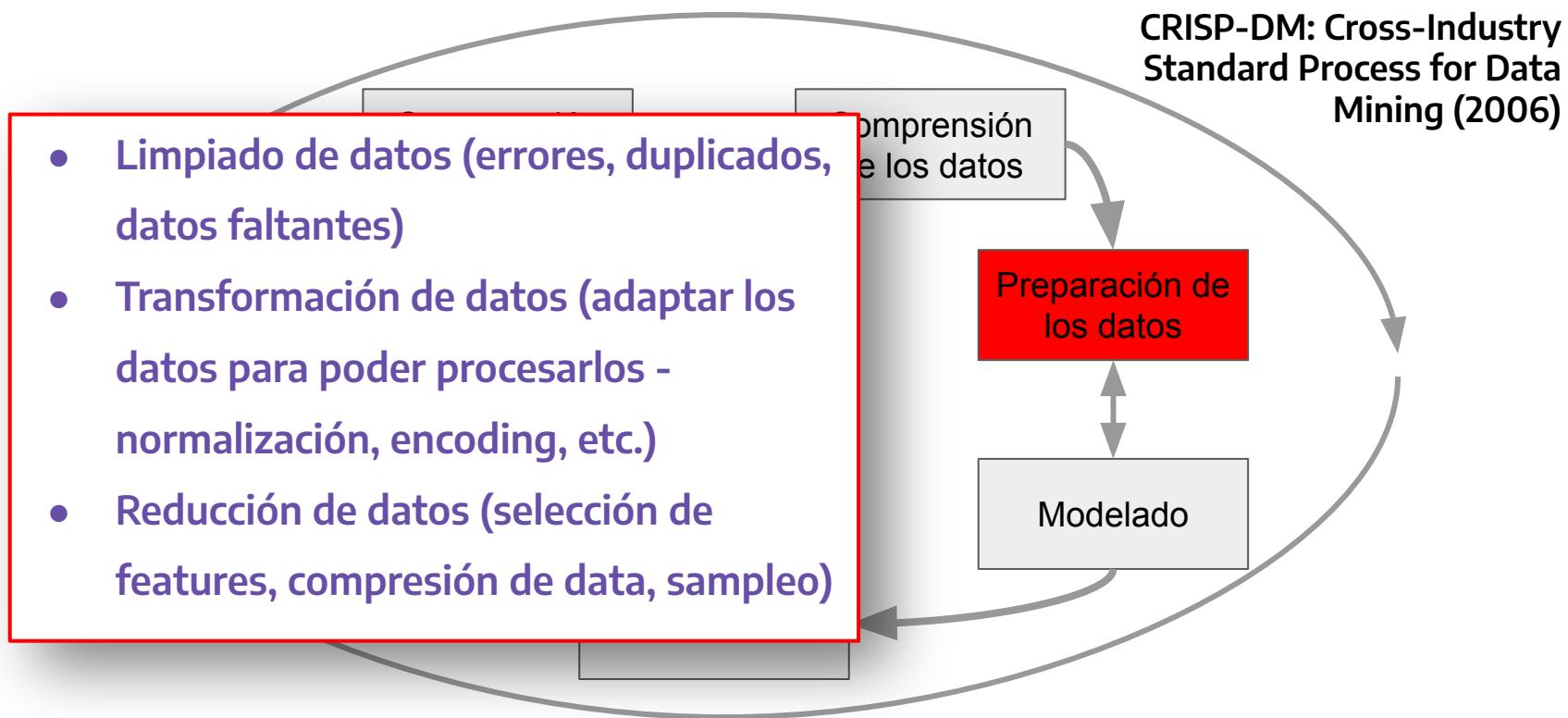
- ¿Qué problema queremos resolver?
- ¿Qué datos podemos conseguir?

- Analizar los datos, explorarlos
- ¿Qué podemos hacer con ellos?
- ¿Qué modelos podemos aplicar?

Evaluación

Modelado

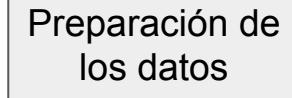
# El proceso de ciencia de datos



# El proceso de ciencia de datos

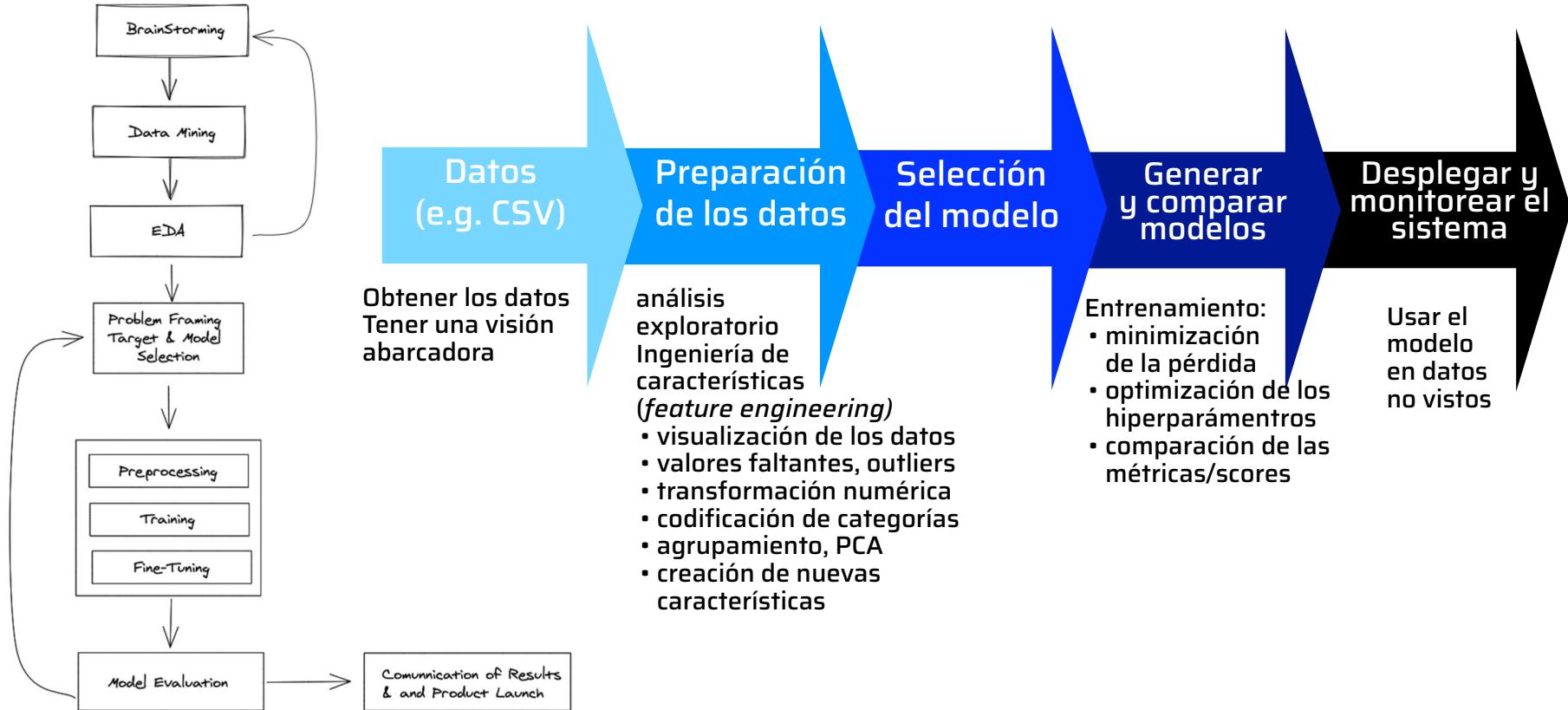


- Precisar la pregunta de interés.
- Definir los objetivos del proyecto.
- Se toman en cuenta las necesidades del negocio.
  - Reuniones en grupos específicos (venta, marketing, ...)
  - Interacciones con administradores de datos.
- Identificar si los datos necesarios para el proyecto están disponibles.
- Se consideran asuntos de ética y privacidad



- Crear un dataset coherente para el análisis posterior.
- Integrar datos de diversas fuentes (bases de datos).
- Controlar la calidad de los datos (faltantes, outliers, ...).

# Pasos del aprendizaje automático





# Una tabla de datos

Variable

Valor

Observación / DataPoint

Dimensión

year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	carrier	flight	tailnum	origin	dest	air_time	distance	hour	minute	time_hour	
0	2013	1	1	517.0	515	2.0	830.0	819	11.0	UA	1545	N14228	EWR	IAH	227.0	1400	5	15	2013-01-01T10:00:00Z
1	2013	1	1	533.0	529	4.0	850.0	830	20.0	UA	1714	N24211	LGA	IAH	227.0	1416	5	29	2013-01-01T10:00:00Z
2	2013	1	1	542.0	540	2.0	923.0	850	33.0	AA	1141	N619AA	JFK	MIA	160.0	1089	5	40	2013-01-01T10:00:00Z
3	2013	1	1	544.0	545	-1.0	1004.0	1022	-18.0	B6	725	N804JB	JFK	BQN	183.0	1576	5	45	2013-01-01T10:00:00Z
4	2013	1	1	554.0	600	-6.0	812.0	837	-25.0	DL	461	N668DN	LGA	ATL	116.0	762	6	0	2013-01-01T11:00:00Z
5	2013	1	1	554.0	558	-4.0	740.0	728	12.0	UA	1696	N39463	EWR	ORD	150.0	719	5	58	2013-01-01T10:00:00Z
6	2013	1	1	555.0	600	-5.0	913.0	854	19.0	B6	507	N516JB	EWR	FLL	158.0	1065	6	0	2013-01-01T11:00:00Z
7	2013	1	1	557.0	600	-3.0	709.0	723	-14.0	EV	5708	N829AS	LGA	IAD	53.0	229	6	0	2013-01-01T11:00:00Z
8	2013	1	1	557.0	600	-3.0	838.0	846	-8.0	B6	79	N593JB	JFK	MCO	140.0	944	6	0	2013-01-01T11:00:00Z
9	2013	1	1	558.0	600	-2.0	753.0	745	8.0	AA	301	N3ALAA	LGA	ORD	138.0	733	6	0	2013-01-01T11:00:00Z
10	2013	1	1	558.0	600	-2.0	849.0	851	-2.0	B6	49	N793JB	JFK	PBI	149.0	1028	6	0	2013-01-01T11:00:00Z
11	2013	1	1	558.0	600	-2.0	853.0	856	-3.0	B6	71	N657JB	JFK	TPA	158.0	1005	6	0	2013-01-01T11:00:00Z
12	2013	1	1	558.0	600	-2.0	924.0	917	7.0	UA	194	N29129	JFK	LAX	345.0	2475	6	0	2013-01-01T11:00:00Z
13	2013	1	1	558.0	600	-2.0	923.0	937	-14.0	UA	1124	N53441	EWR	SFO	361.0	2565	6	0	2013-01-01T11:00:00Z
14	2013	1	1	559.0	600	-1.0	941.0	910	31.0	AA	707	N3DUAA	LGA	DFW	257.0	1389	6	0	2013-01-01T11:00:00Z
15	2013	1	1	559.0	559	0.0	702.0	706	-4.0	B6	1806	N708JB	JFK	BOS	44.0	187	5	59	2013-01-01T10:00:00Z
16	2013	1	1	559.0	600	-1.0	854.0	902	-8.0	UA	1187	N76515	EWR	LAS	337.0	2227	6	0	2013-01-01T11:00:00Z
17	2013	1	1	600.0	600	0.0	851.0	858	-7.0	B6	371	N595JB	LGA	FLL	152.0	1076	6	0	2013-01-01T11:00:00Z
18	2013	1	1	600.0	600	0.0	837.0	825	12.0	MQ	4650	N542MQ	LGA	ATL	134.0	762	6	0	2013-01-01T11:00:00Z
19	2013	1	1	601.0	600	1.0	844.0	850	-6.0	B6	343	N644JB	EWR	PBI	147.0	1023	6	0	2013-01-01T11:00:00Z
20	2013	1	1	602.0	610	-8.0	812.0	820	-8.0	DL	1919	N971DL	LGA	MSP	170.0	1020	6	10	2013-01-01T11:00:00Z
21	2013	1	1	602.0	605	-3.0	821.0	805	16.0	MQ	4401	N730MQ	LGA	DTW	105.0	502	6	5	2013-01-01T11:00:00Z
22	2013	1	1	606.0	610	-4.0	858.0	910	-12.0	AA	1895	N633AA	EWR	MIA	152.0	1085	6	10	2013-01-01T11:00:00Z
23	2013	1	1	606.0	610	-4.0	837.0	845	-8.0	DL	1743	N3739P	JFK	ATL	128.0	760	6	10	2013-01-01T11:00:00Z
24	2013	1	1	607.0	607	0.0	858.0	915	-17.0	UA	1077	N53442	EWR	MIA	157.0	1085	6	7	2013-01-01T11:00:00Z
25	2013	1	1	608.0	600	8.0	807.0	735	32.0	MQ	3768	N9EAMQ	EWR	ORD	139.0	719	6	0	2013-01-01T11:00:00Z
26	2013	1	1	611.0	600	11.0	945.0	931	14.0	UA	303	N532UA	JFK	SFO	366.0	2586	6	0	2013-01-01T11:00:00Z
27	2013	1	1	613.0	610	3.0	925.0	921	4.0	B6	135	N635JB	JFK	RSW	175.0	1074	6	10	2013-01-01T11:00:00Z
28	2013	1	1	615.0	615	0.0	1039.0	1100	-21.0	B6	709	N794JB	JFK	SJU	182.0	1598	6	15	2013-01-01T11:00:00Z
29	2013	1	1	615.0	615	0.0	833.0	842	-9.0	DL	575	N326NB	EWR	ATL	120.0	746	6	15	2013-01-01T11:00:00Z

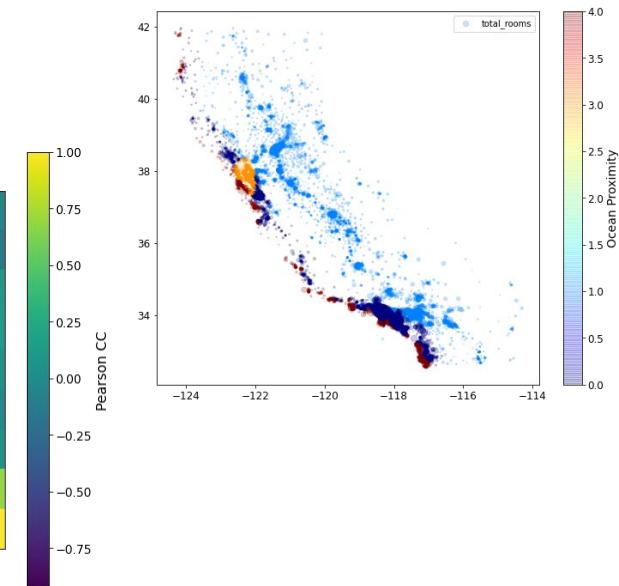
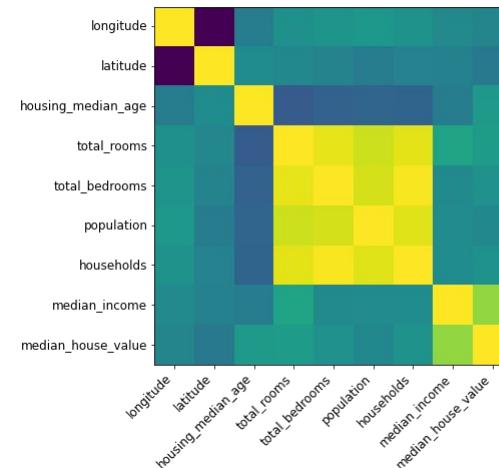
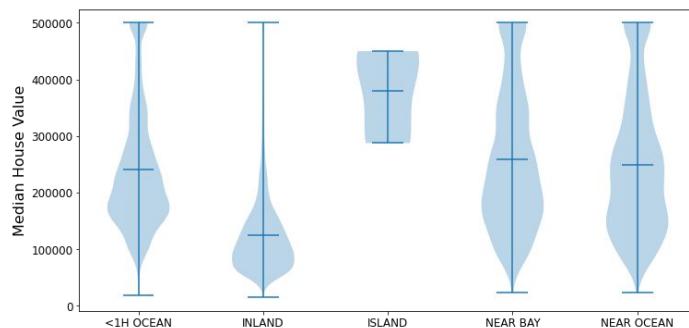
# Preparación de los datos

- Recolectar los datos (aquí: bajar de la red, usar conjuntos nativos de [sklearn](#), generación de datos simulados)
- Análisis exploratorio: contenidos, integridad, correlaciones, etc. (atributos de [pandas](#))
  - contenido:

`head, columns, info, unique, value_counts, describe, sort`

- distribuciones de probabilidad y correlaciones

`corr, hist, violin plot, scatter, scatter_matrix`



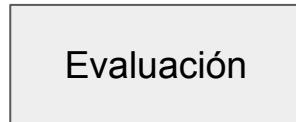
# Preparación de los datos

- Recolectar los datos (aquí: bajar de la red, usar conjuntos nativos de `sklearn`, generación de datos simulados)
- Análisis exploratorio: contenidos, integridad, correlaciones, etc.
  - contenido:  
`head, columns, info, unique, value_counts, describe, sort` (atributos de `pandas`)
    - distribuciones de probabilidad y correlaciones  
`corr, hist, violin plot, scatter, scatter_matrix`
    - otras visualizaciones y análisis
- Preprocesamiento
  - limpieza de los datos
  - codear variables categóricas
  - Ingeniería de características (*feature engineering*)
    - normalizar
    - construcción de nuevas variables
    - transformación (e.g. proyección, PCA)
  - clustering, reducción dimensional
  - supervisado: definir un target `v` (que queremos predecir) y separarlo de los datos `x`

# El proceso de ciencia de datos



- Extracción de patrones útiles de los datos.
- Involucra modelos sencillos y/o de machine learning.
- Experimentar con diversos algoritmos.
- Se identifican nuevos problemas con los datos, o con su preparación

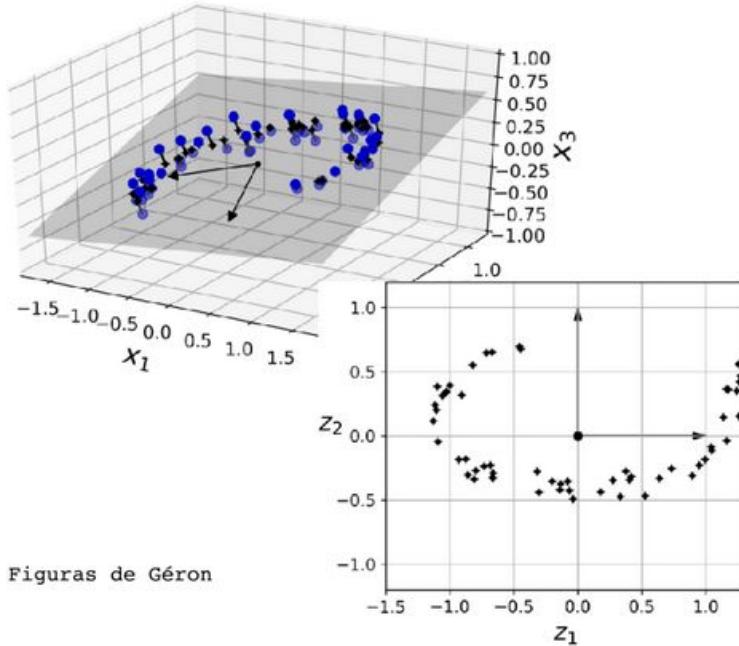


- Evaluar el modelo en un contexto más amplio (p.ej., con datos que no se hayan visto antes).
- Evaluar si cumple los objetivos del proyecto.
- Llevar la solución generada al ambiente del negocio.
- Planear la integración con soluciones existentes
- Planear la evaluación periódica del desempeño del modelo.



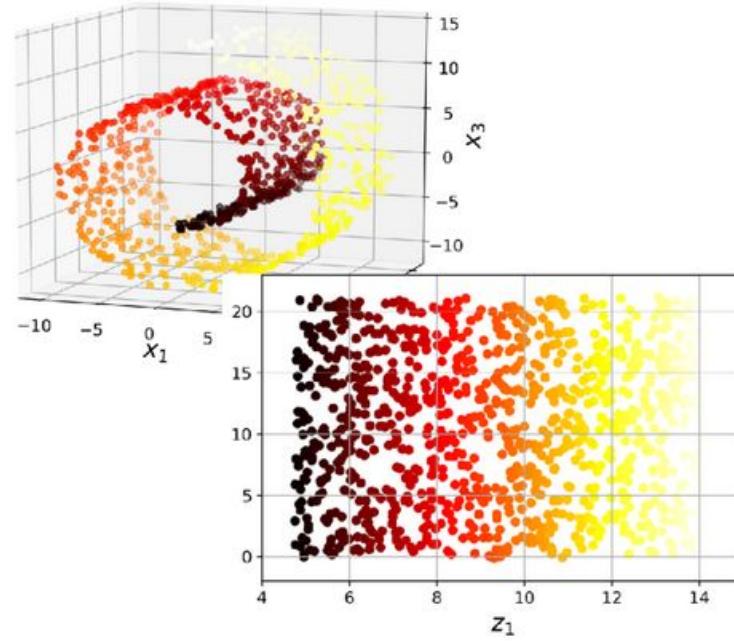
# Reducción dimensional

- Proyección

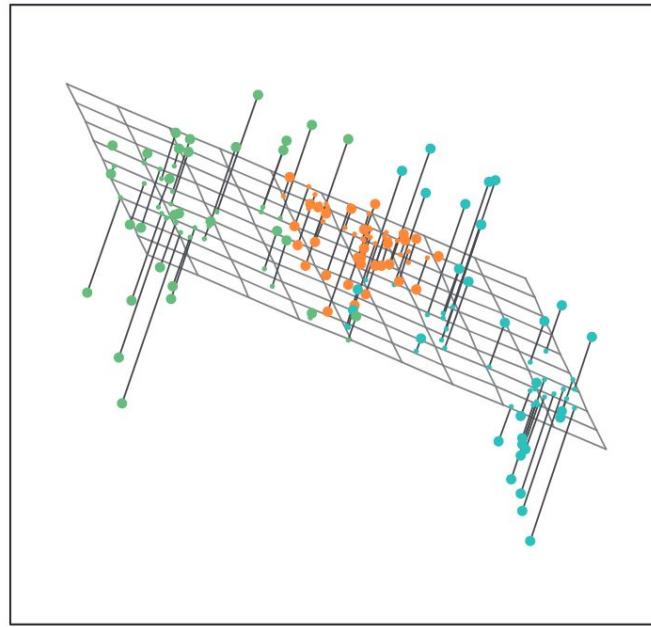


Figuras de Géron

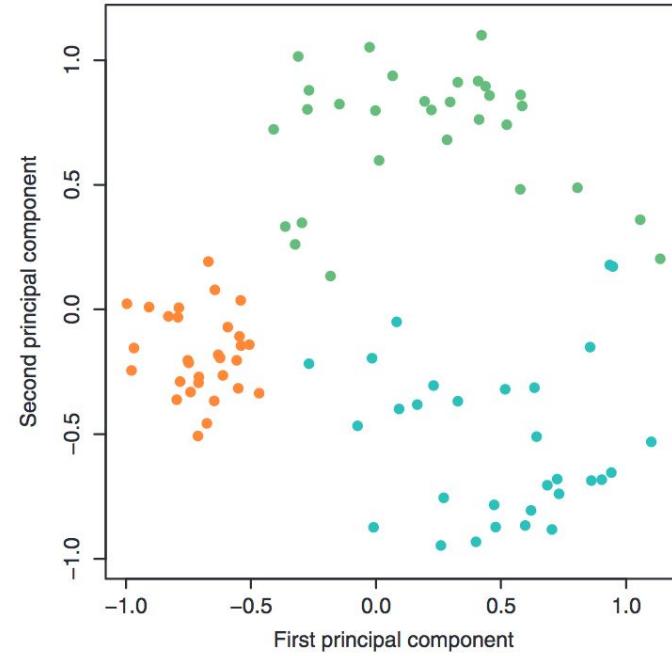
- Manifold learning



# PCA y reducción de la dimensionalidad



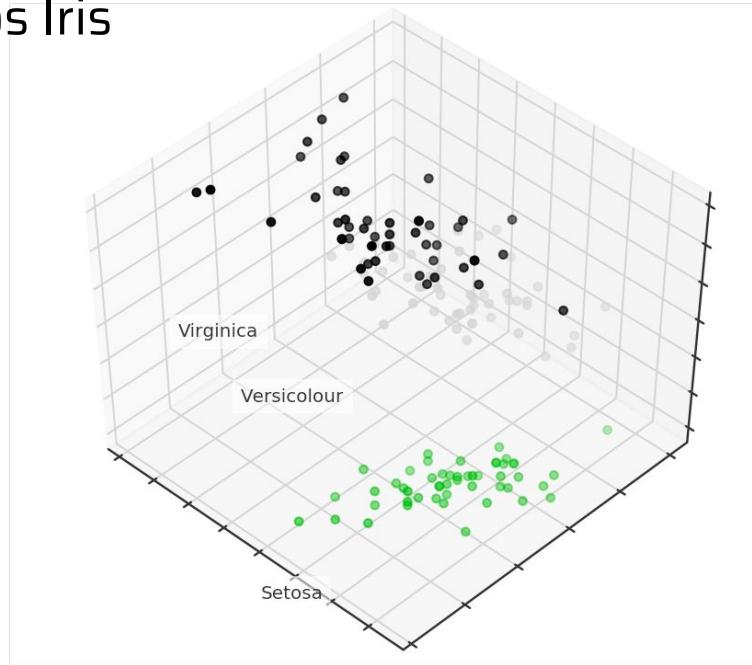
Plano en el cual hay menos dispersión - información!



Elegir 2 coordinates en el plano y deshacerse de la 3a

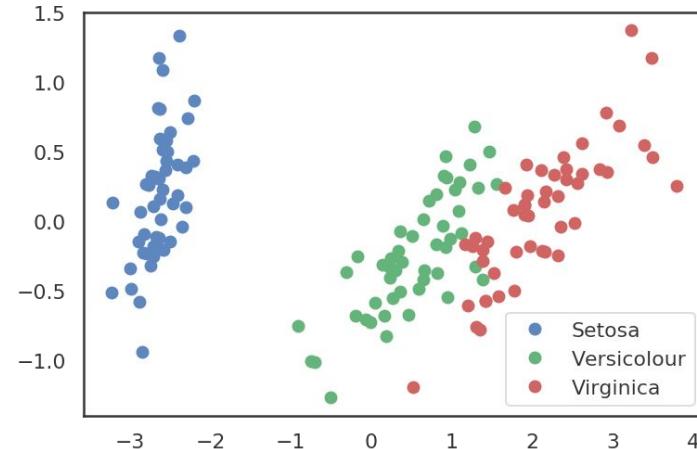
# PCA y reducción de la dimensionalidad

## Datos Iris



1a componente: 92.46% de la varianza  
2a componente: 5.31%

Gráfica usando las dos primeras PCA

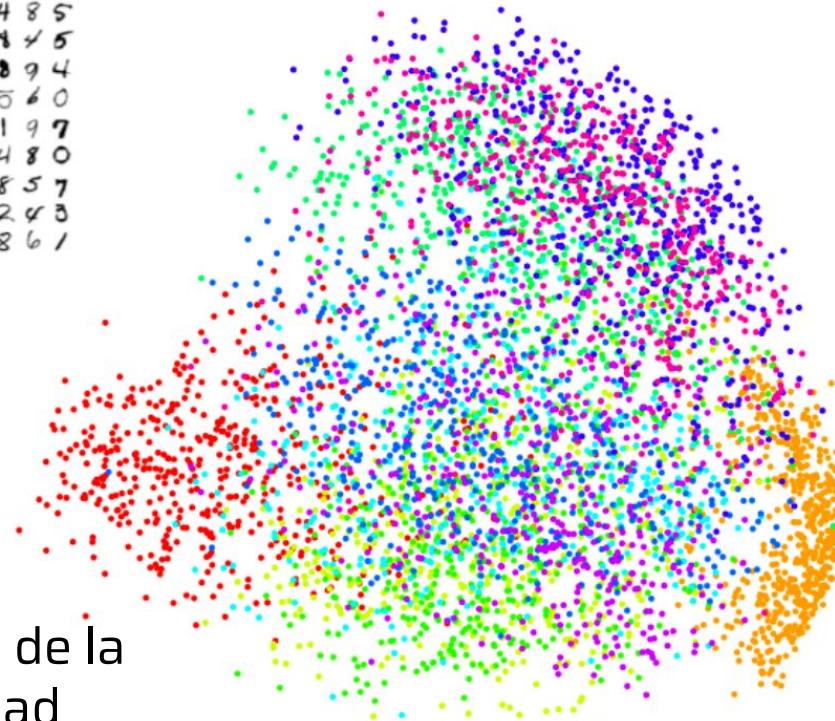


Las 3 clases se separan mucho mejor que usando solo 1 variable o eligiendo 2 de las originales

# Ejemplo: reducción dimensional en MNIST

```
3 6 8 1 7 9 6 6 9 1  
6 7 5 7 8 6 3 4 8 5  
2 1 7 9 7 1 2 8 4 5  
4 8 1 9 0 1 8 8 9 4  
7 6 1 8 6 4 1 5 6 0  
7 5 9 2 6 5 8 1 9 7  
1 2 2 2 2 3 4 4 8 0  
0 2 3 8 0 7 3 8 5 7  
0 1 4 6 4 6 0 2 4 3  
7 1 2 8 7 6 9 8 6 1
```

Cada dígito tiene más de 1000 dimensiones



Posición de cada instancia en el espacio de las dos primeras PCA

Reducción brutal de la dimensionalidad

# Modelado de datos

## Objetivos

- **Cuantificar una relación.** Es decir, ponerle números a eso: "el precio de las casas es de X USD por metro cuadrado").
- **Explorar** los datos. Muchas veces, necesitamos quitar los patrones más obvios para poder detectar cosas más sutiles. En este caso, el patrón obvio es la dependencia con la superficie. ¿Habrá algún efecto secundario con, por ejemplo, el estado de la casa, o la cantidad de baños?
- **Resumir** la información para transmitirla mejor. Un excelente complemento al gráfico de arriba es dar los números de los parámetros (ver abajo), que resultan una descripción compacta de los datos.
- **Predecir** el valor de la variable target para una propiedad que no hemos observado.

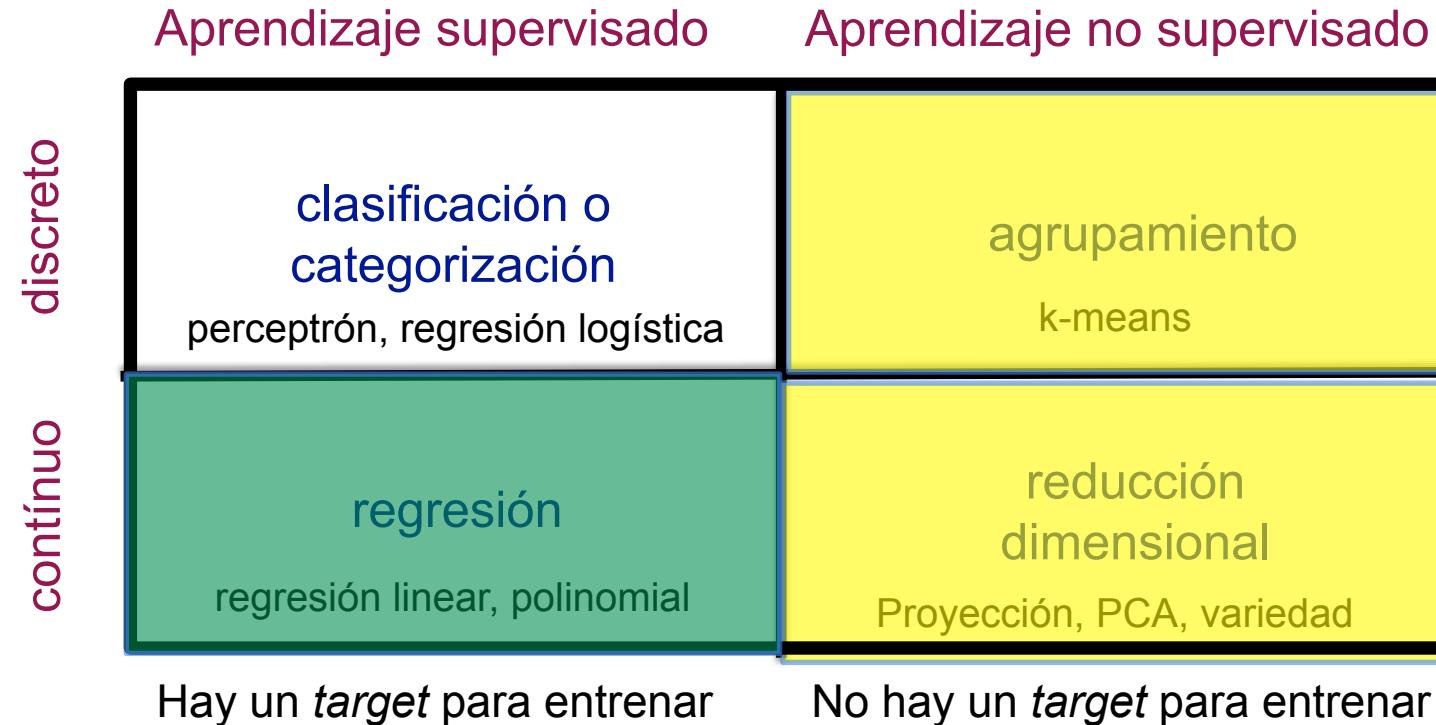


**Argentina  
programa  
4.0**

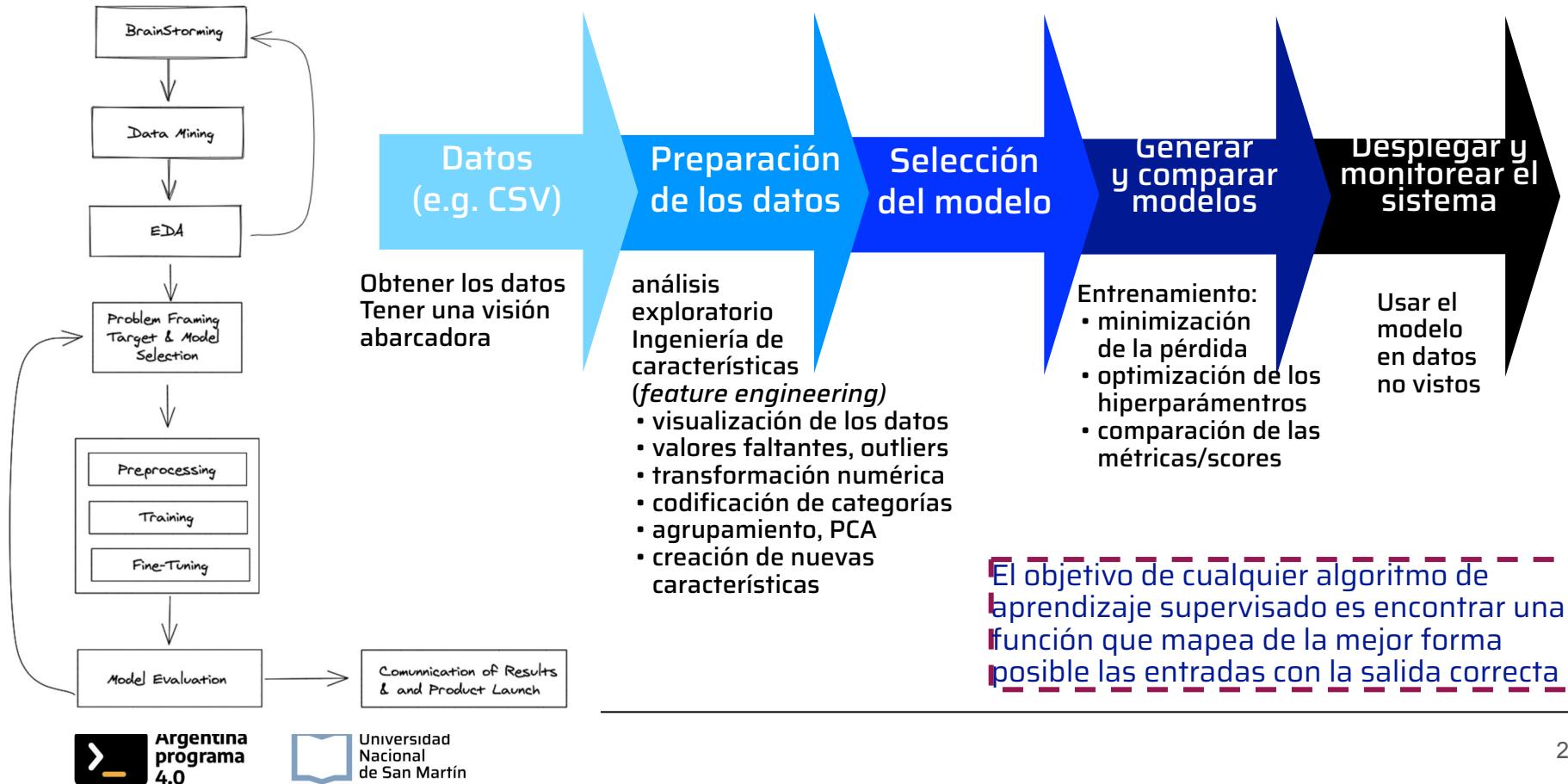
---

# Regresión

# Tipos de datos y problemas

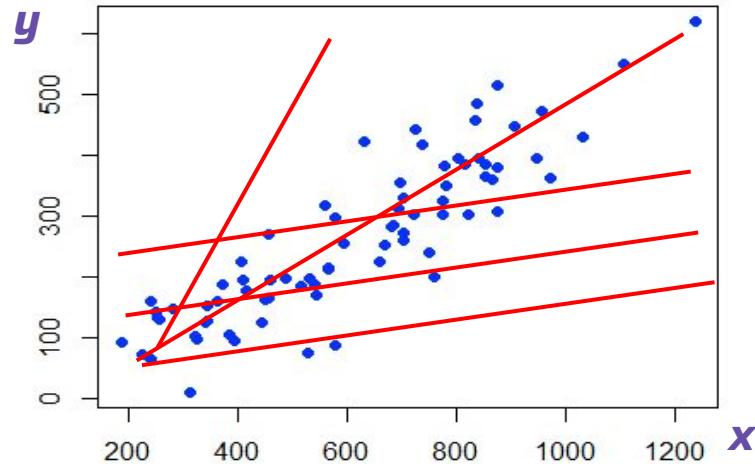


# Pasos del aprendizaje automático



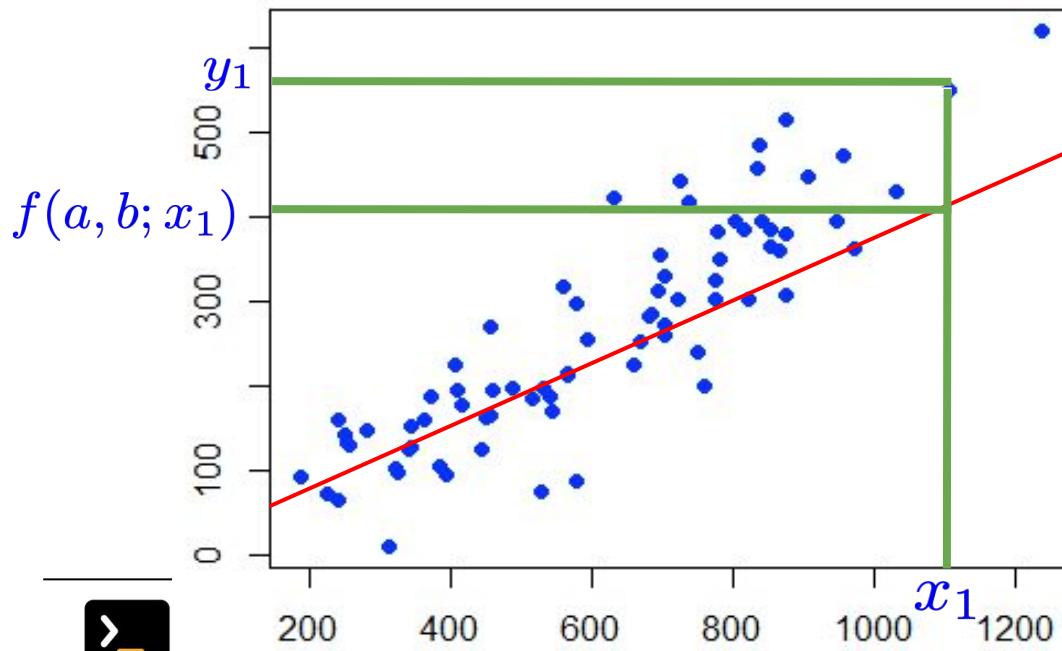
# Datos y modelos

- Normalmente los datos son discretos:
  - Ejemplo: pares de números  
pueden visualizarse como un gráfico de dispersión
- Frecuentemente los modelos vienen dados por funciones continuas
  - Ej.: función continua de una variable  $f(x)$
  - Puede representarse en un gráfico  $y = f(x)$
  - Ejemplo de modelo:  $y = a x + b$ 
    - variable
    - parámetros
  - Parámetros determinados a partir de los datos



# Ajustando una función a los datos

- Encontrar un modelo (función) que represente (aproximadamente) los datos
- Minimizar la distancia de un modelo a los datos



En este caso

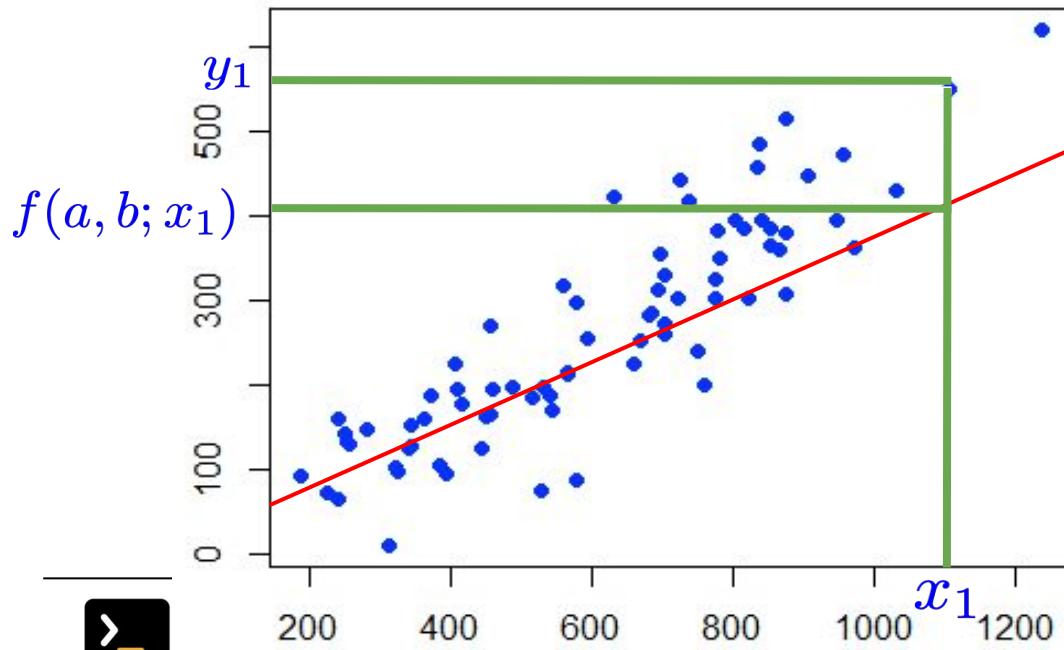
$$f(a, b; x) = ax + b$$

Distancia entre la previsión del modelo y el dato:

$$\begin{aligned} & f(a, b; x_1) - y_1 \\ &= ax_1 + b - y_1 \end{aligned}$$

# Ajustando un modelo a los datos

- Encontrar un modelo (función) que represente (aproximadamente) los datos
- Minimizar la distancia de un modelo a los datos



Distancia cuadrática entre todos los puntos y la predicción:

$$\begin{aligned} & (f(a, b; x_1) - y_1)^2 + \\ & (f(a, b; x_2) - y_2)^2 + \\ & (f(a, b; x_3) - y_3)^2 + \\ & \quad \dots \\ & = D(a, b) \end{aligned}$$

# Ajustando un modelo a los datos

- Encontrar un modelo (función) que represente (aproximadamente) los datos
- Minimizar la distancia de un modelo a los datos

El modelo que mejor representa los datos (dentro de esa categoría de modelos) es el que minimiza la función

$$D(a, b)$$

Una vez que obtenemos  $a$  y  $b$ , podemos hacer predicciones para nuevos valores de  $x$

En el caso simple de la función  $y = a x + b$  es muy fácil encontrar  $a$  y  $b$  a partir del conjunto de todos los  $x_i$  e  $y_i$  (derivadas, etc.) ver expresión en el notebook

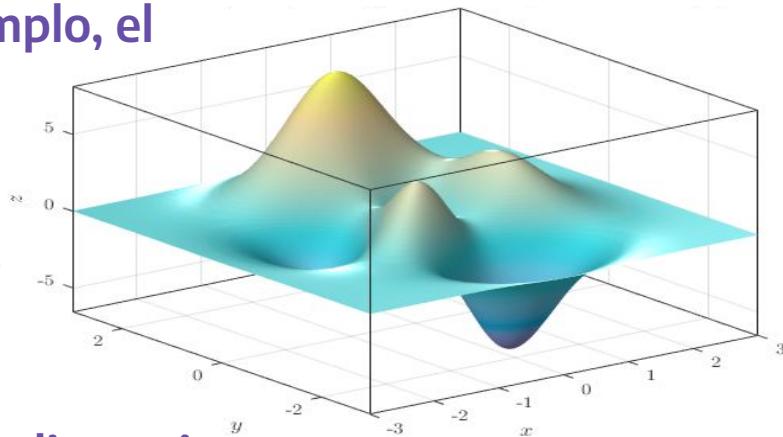
Distancia cuadrática entre todos los puntos y la predicción:

$$\begin{aligned} & (f(a, b; x_1) - y_1)^2 + \\ & (f(a, b; x_2) - y_2)^2 + \\ & (f(a, b; x_3) - y_3)^2 + \\ & \dots \\ & = D(a, b) \end{aligned}$$

# Recordando: Maximización y minimización

Minimizar (o maximizar) una función (por ejemplo, el beneficio o la distancia cuadrática):

- Puntos extremos
- Pendientes nulas en todas las direcciones (gradiente)
- Muy utilizado en aprendizaje automático
- Encontrar máximos y mínimos en muchas dimensiones puede ser muy complicado
- En este caso de ajuste de función, las dimensiones son los parámetros del modelo



# Ajustando un modelo a los datos

- Encontrar un modelo (función) que represente (aproximadamente) los datos
- Minimizar la distancia de un modelo a los datos

El modelo que mejor representa los datos (dentro de esa categoría de modelos) es el que minimiza

$$D(a, b)$$

Una vez que obtenemos  $a$  y  $b$ , podemos hacer predicciones para nuevos valores de  $x$

En el caso simple de la función  $y = a x + b$  es muy fácil encontrar  $a$  y  $b$  a partir del conjunto de todos los  $x_i$  e  $y_i$  (derivadas, etc.) ver expresión en el notebook

En este caso en particular hay una solución exacta y su expresión es:

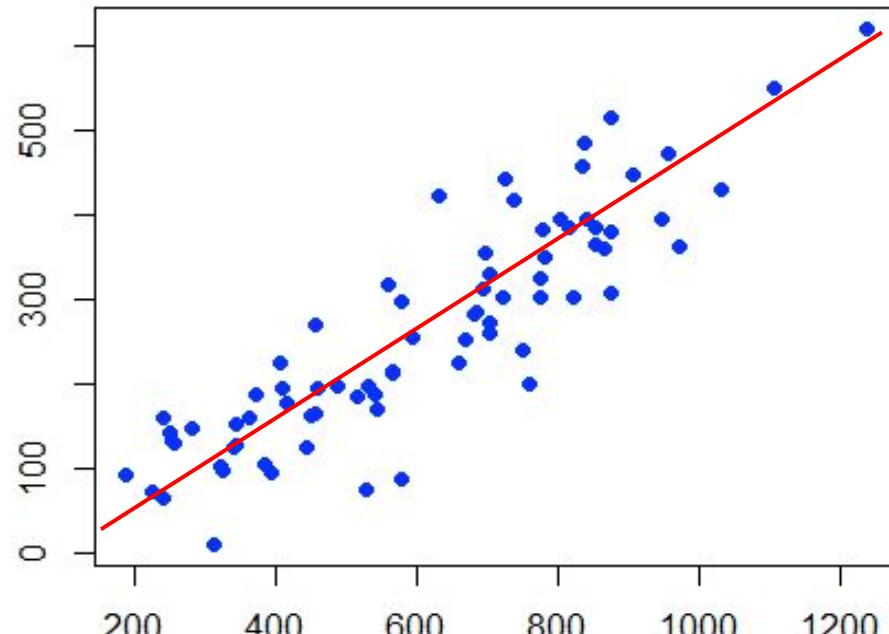
$$a = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$b = \bar{y} - a\bar{x}$$

Ver ejemplo en el notebook

# ¡Aprendiendo de los datos!

- A partir de los datos, obtuvimos un modelo que los representa: el que tiene los parámetros que minimizan la distancia cuadrática promedio a los datos (por eso se llama de "método de mínimos cuadrados")
- El modelo generaliza una relación y permite hacer predicciones



- Se puede pensar que el modelo se entrenó con los datos (fiteando) y ahora puede predecir un valor de  $y$  para cualquier  $x$
- ¡Un algoritmo y un ejemplo simple de aprendizaje automático a partir de datos!

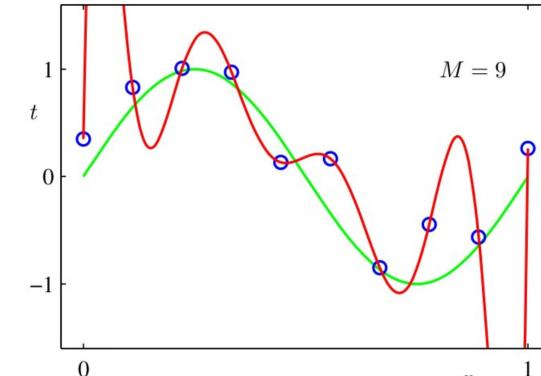
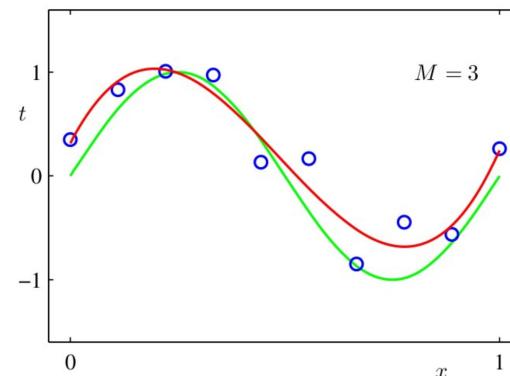
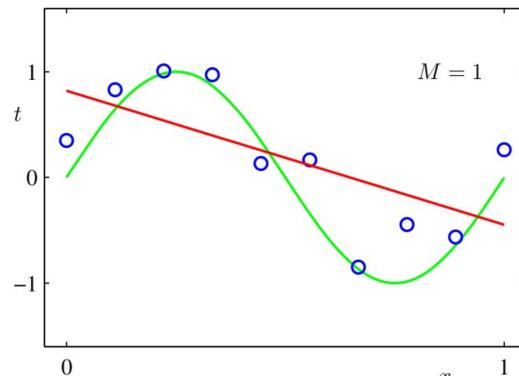
# ¿Cómo esto se generaliza?

- Aquí el modelo era lineal,  $y = ax + b$  (por eso se le llama **regresión lineal**)
- Podemos tener funciones más complejas, como cuadrática, polinómica, etc.

Vector de  
coeficientes

$$y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M$$

- En general tienen más parámetros (= más libertad para ajustar los datos)
- ¡Hay que minimizar en muchas dimensiones!



# Regresión lineal, mínimos cuadrados

- Uno de los algoritmos de aprendizaje automático más simples
  - Minimizar la distancia cuadrática promedio entre los datos y un modelo (MSE)

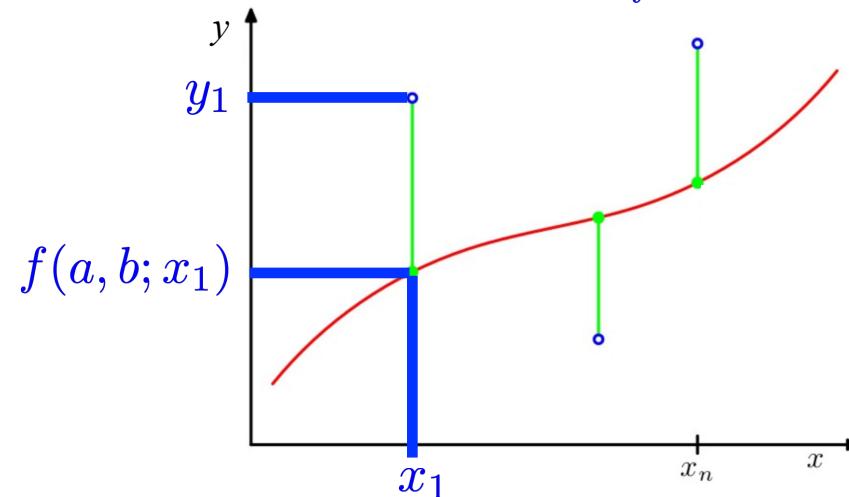
$$D^2(a, b; \{y\}, \{x\}) = \sum_i (f(a, b; x_i) - y_i)^2$$

- Ejemplo: relación lineal  $y = a x + b$
- Cálculo (derivadas) + matrices (inversa, transpuesta):  
< 10 líneas de código, encuentra  $a$  y  $b$  (1 línea en sklearn)
- Eso es una máquina aprendiendo de los datos!

# Regresión lineal, mínimos cuadrados

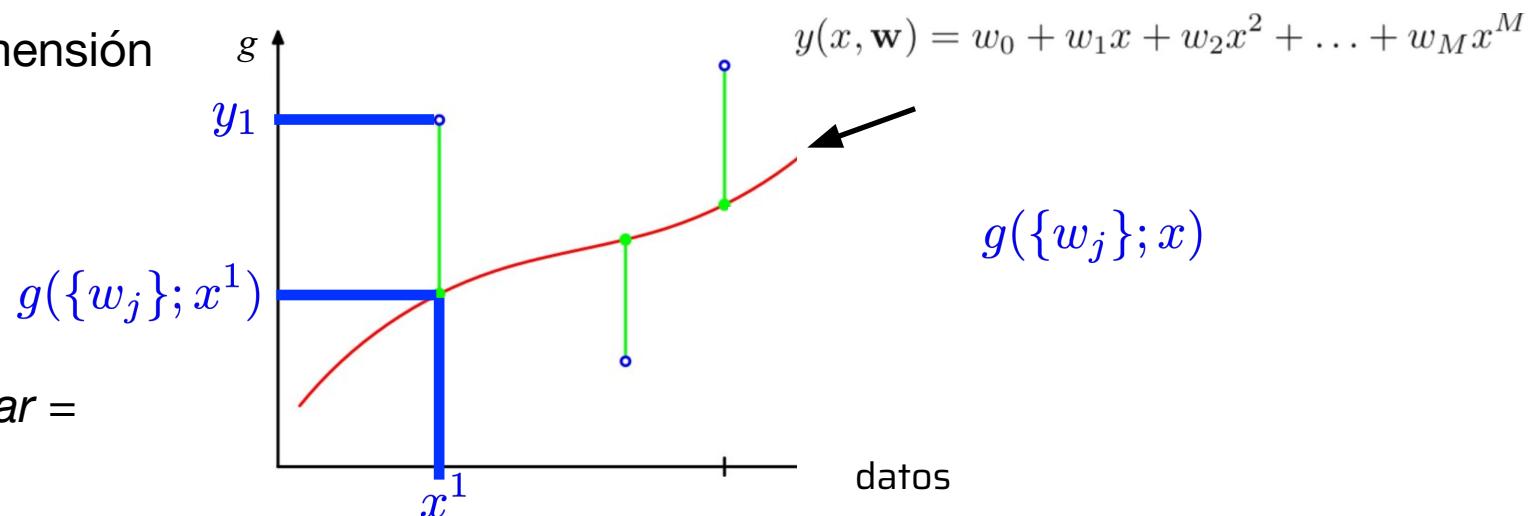
- Uno de los algoritmos de aprendizaje automático más simples
  - Minimizar la distancia cuadrática promedio entre los datos y un modelo (MSE)

$$D^2(a, b; \{y\}, \{x\}) = \sum_i (f(a, b; x_i) - y_i)^2$$



# Ejemplo: regresión lineal/polinomial

En una dimensión



Aprender = *fitear* =  
minimizar la  
pérdida

(aquí, el *MSE*)  $MSE(\{w_j\}) = D(\{w_j\}; \{x_i, y\}) = \sum (g(w_j; \{x_i\}) - y)^2$

todos los pesos  
(dimensiones, polinómico)

características/datos

- La **forma** de  $g$  depende de los
- valores de los pesos  $w_j$

El objetivo de cualquier algoritmo de aprendizaje supervisado es encontrar una función que mapea de la mejor forma posible las entradas con la salida correcta

# Aprendizaje y función pérdida

- Aprendizaje supervisado necesita datos con el resultado esperado (*target*)
- Compara la salida de un modelo (predicción) con el valor esperado (target)
- Ejemplo: en un problema de **regression**, usar el *the Mean Squared Error*:

$$MSE(\{w_j\}) = D(\{w_j\}; \{x_i, y\}) = \sum \text{suma sobre todos los datos} (g(w_j; \{x_i\}) - y)^2$$

parámetros (pesos)

previsión (modelo) (ej: lineal)

sumar sobre todos los datos

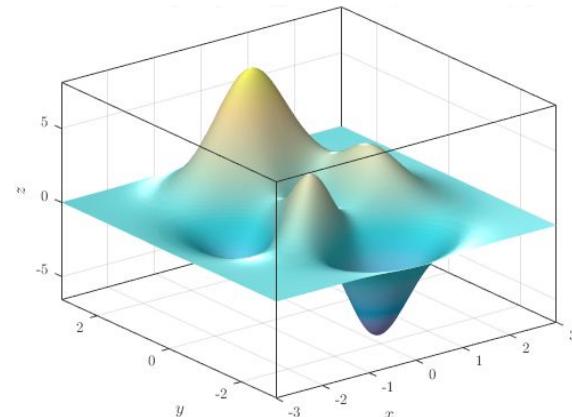
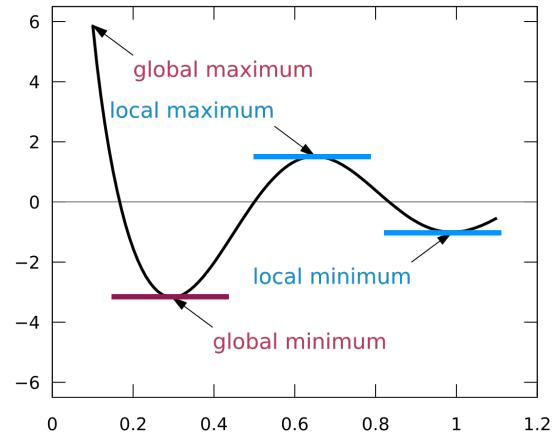
# Función de pérdida (*loss function*)

El objetivo de cualquier algoritmo de aprendizaje supervisado es encontrar una función que mapea de la mejor forma posible las entradas con la salida correcta

- General para aprendizaje automático (ML) supervisado (y a veces no supervisado también):
    - “aprender es minimizar la función de pérdida”
    - Se necesitan los datos etiquetados: valores/números o clasificaciones correctas (*target*),  $t$
  - Para regresión, en general, error cuadrático medio, MSE
  - Para clasificación, en general *cross entropy*
  - Muchos métodos de minimización
  - Agregar una regularización
- 
- Durante el entrenamiento, los pares entrada-salida,  $(\mathbf{X}, \mathbf{t})$  son fijos. Los parámetros varían y se calcula una función de pérdida.
  - Cuando se evalúa un modelo, los *parámetros* son fijos, las entradas varían (y el *target* puede ser desconocido). El proceso de predicción termina con la salida (no hay función pérdida)

# Minimización y maximización

- En general, *aprender pasa por maximizar o minimizar una función!*
  - Minimizar una función de pérdida (*loss function*)
  - Maximizar un score (o el lucro):
  - Primer derivada nula
  - Segunda derivada positiva (o negativa)
- Muy complejo en muchas dimensiones y si hay muchos mínimos o máximos locales:
  - Riqueza de métodos numéricos que **no** vamos a explorar



# Aprendizaje Supervisado

## Entrenamiento (y validación)

(después de definir y pre-procesar los datos, elegir método, hiperparámetros, etc.)

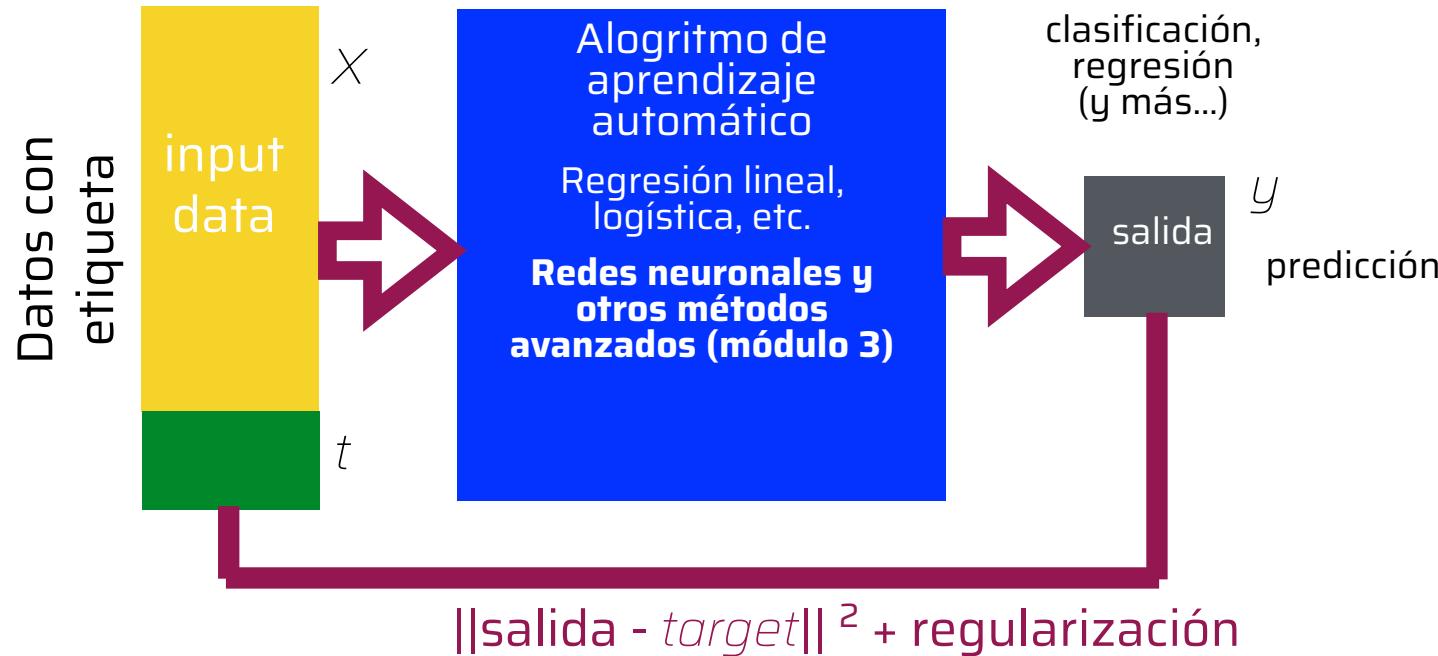


función de pérdida (*loss*): comparación de la salida con el target

# Aprendizaje Supervisado

## Entrenamiento (y validación)

(después de definir y pre-procesar los datos, elegir método, hiperparámetros, etc.)



# Aprendizaje Supervisado

## Entrenamiento (y validación)

(después de definir y pre-procesar los datos, elegir método, hiperparámetros, etc.)



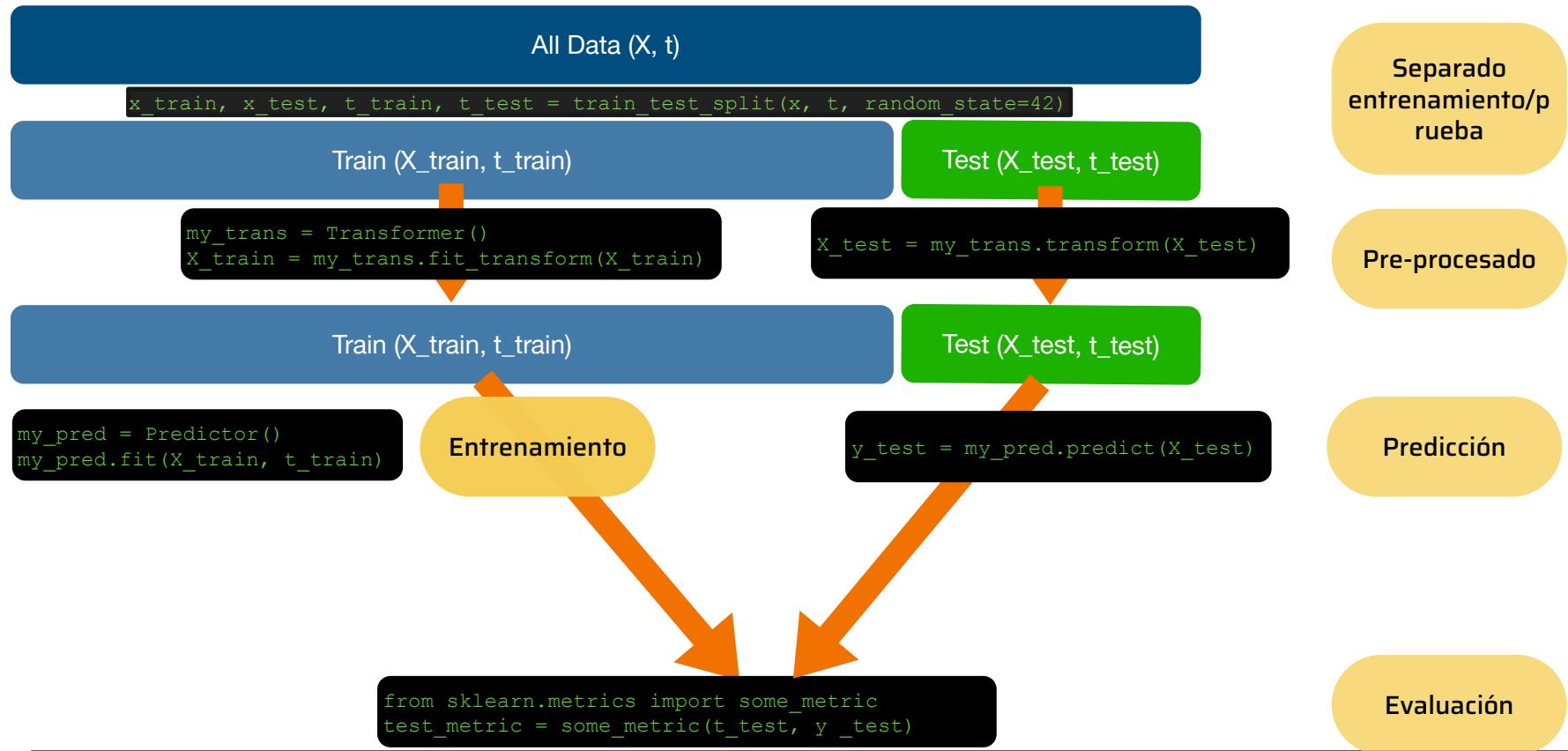
En ese proceso: evitar sobreajuste, mínimos locales, estimar los errores (e.g. validación cruzada: separación en entrenamiento y validación), usar metricas (e.g. AUC), elegir el *working point*, ajustar hiperparámetros (probar con distintos métodos)

# Aprendizaje Supervisado

## Uso



# Aprendizaje supervisado en la práctica

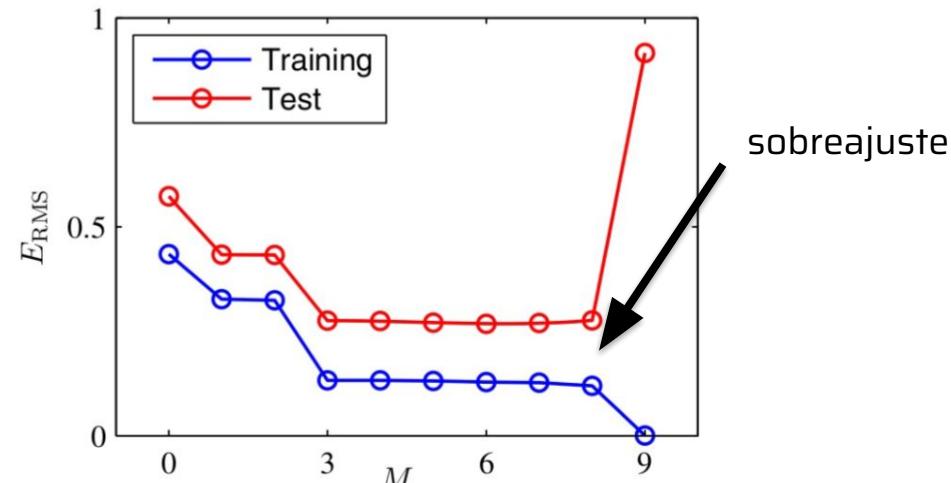
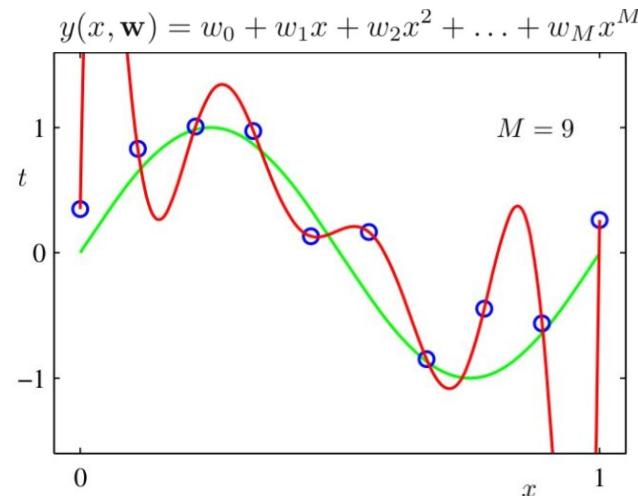


# Parámetros e hiperparámetros

- Parámetros: se ajustan (**fit**) en el aprendizaje
  - minimización en relación a esos parámetros
  - ejemplo: coeficientes en un ajuste lineal o polinomial, pesos  $w_i$  en el argumento de la función logística, centros de los grupos, etc.
- Hiperparámetros: definen el **modelo**
  - Ejemplos: orden del polinómio, uso de dimensiones, número de grupos, número de PCA
  - Más hiperparámetros = más complejidad
    - posibilidad de ajustar mejor los datos
    - riesgo de sobreajuste
    - ejemplo: orden del polinómio, coeficiente de regularización

# Hiperparámetros y validación cruzada

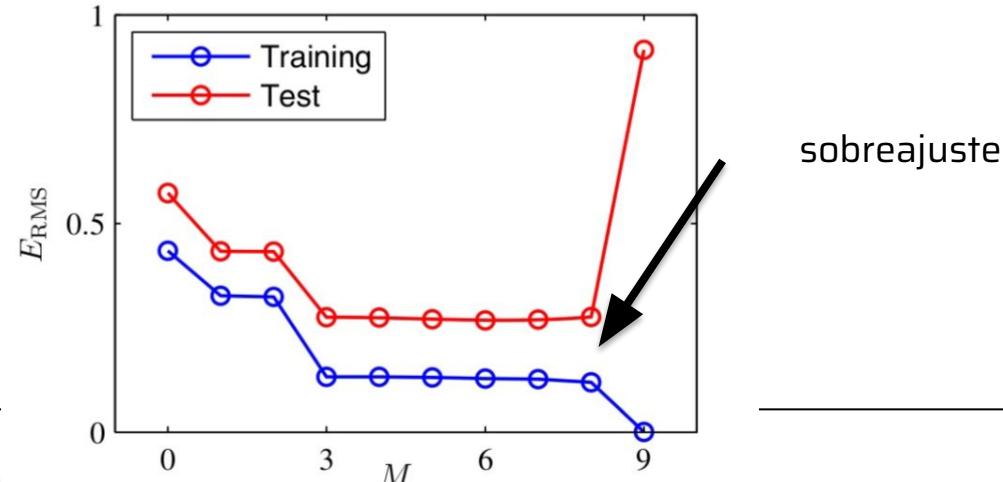
- Como elegir los hiperparámetros?
- Mejorar la pérdida (*loss*), e.g., la MSE
- Evitar sobreajuste (aprender los datos): mirar en los datos de entrenamiento y de prueba



- Y si al ajustar los *hiperparametros* también estamos sobreajustando para una combinación particular de entrenamiento + prueba?
- Como validar, de forma independiente, la elección de los hiperparámetros?

# Hiperparámetros y validación cruzada

- Y si al ajustar los *hiperparametros* también estamos sobre ajustando para una combinación particular de entrenamiento + prueba?
- Como validar, de forma independiente, la elección de los hiperparámetros?
- Validación cruzada: separar los datos etiquetados (con *target*) en un conjunto de entrenamiento y de validación (e.g. 80% + 20%)
- La pérdida (*loss*) en el conjunto de validación debe ser comparable a la del conjunto de entrenamiento. Si no, estamos sobreajustando

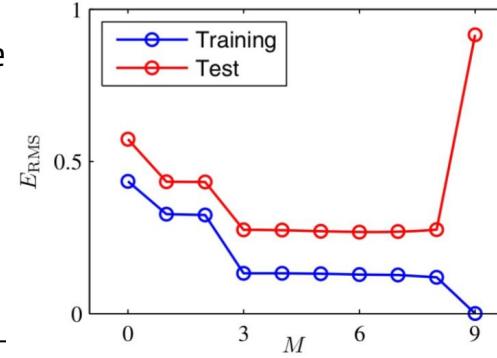


# Resumen: como evitar el sobreajuste

- Repetir la separación: entrenamiento + validación, seleccionando aleatoriamente las sub-muestras
- además de evitar el sobreajuste de los hiperparámetros la CV permite **estimar errores!**

- **Además: usar regularización**

- Encontrar un balance entre el número de parámetros (por ejemplo, los  $w_j$ ) y el número de datos
- Eso es simple para un ajuste polinomial en 1D, pero no para redes neuronales complejas y grandes conjuntos de datos
- Validación cruzada: separar los datos etiquetados (con  $\text{target}$ ) en un conjunto de entrenamiento y de validación (e.g. 80% + 20%)
- La pérdida ( $\text{loss}$ ) en el conjunto de validación debe ser comparable estamos sobreajustando



o. Si no,

# Regularización *ridge*

## ▼ Función de error modificada

El hecho de que los coeficientes aumentan abruptamente cuando empezamos a sobreajustar nos da una idea de incluir una penalización para valores grandes de los parámetros. Una forma de implementar eso es agregar un nuevo término a la función error:

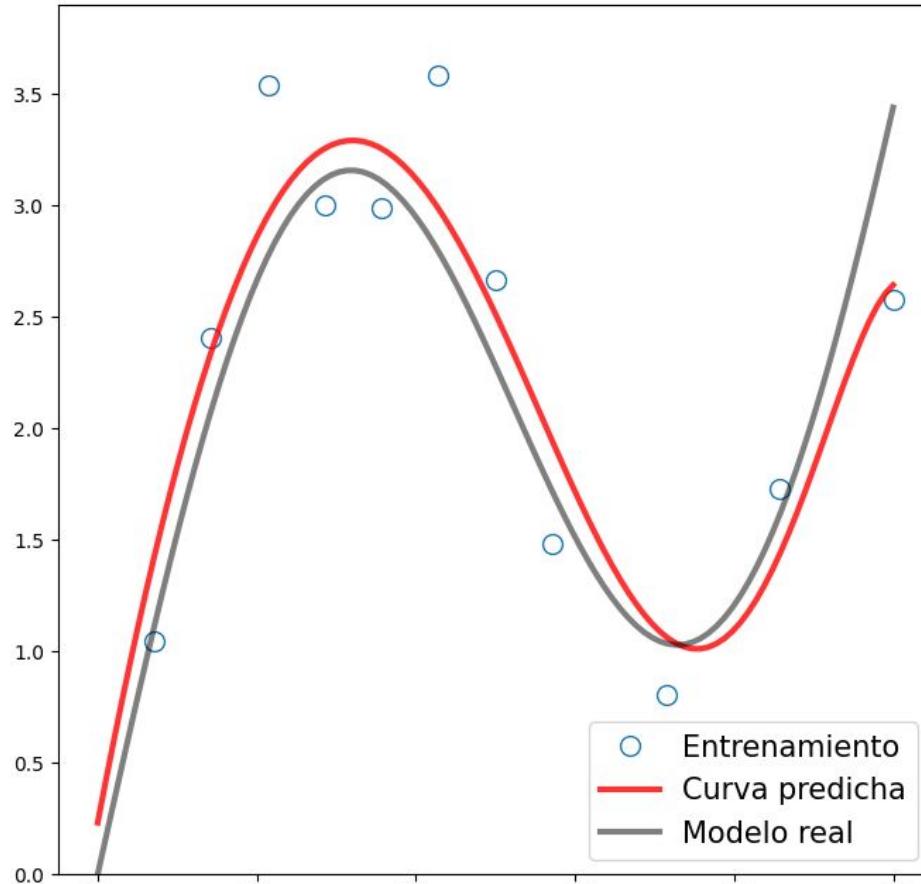
$$E_{\text{ridge}}(\boldsymbol{\omega}; \lambda) = \frac{1}{2} \sum_{i=1}^N \{y(x_i, \boldsymbol{\omega}) - t_i\}^2 + \frac{\lambda}{2} \sum_{i=1}^M \omega_i^2 .$$

El nuevo término es llamado de término de regularización (o penalización). La parte que multiplica  $\lambda/2$  es el cuadrado de la norma del vector de parámetros,

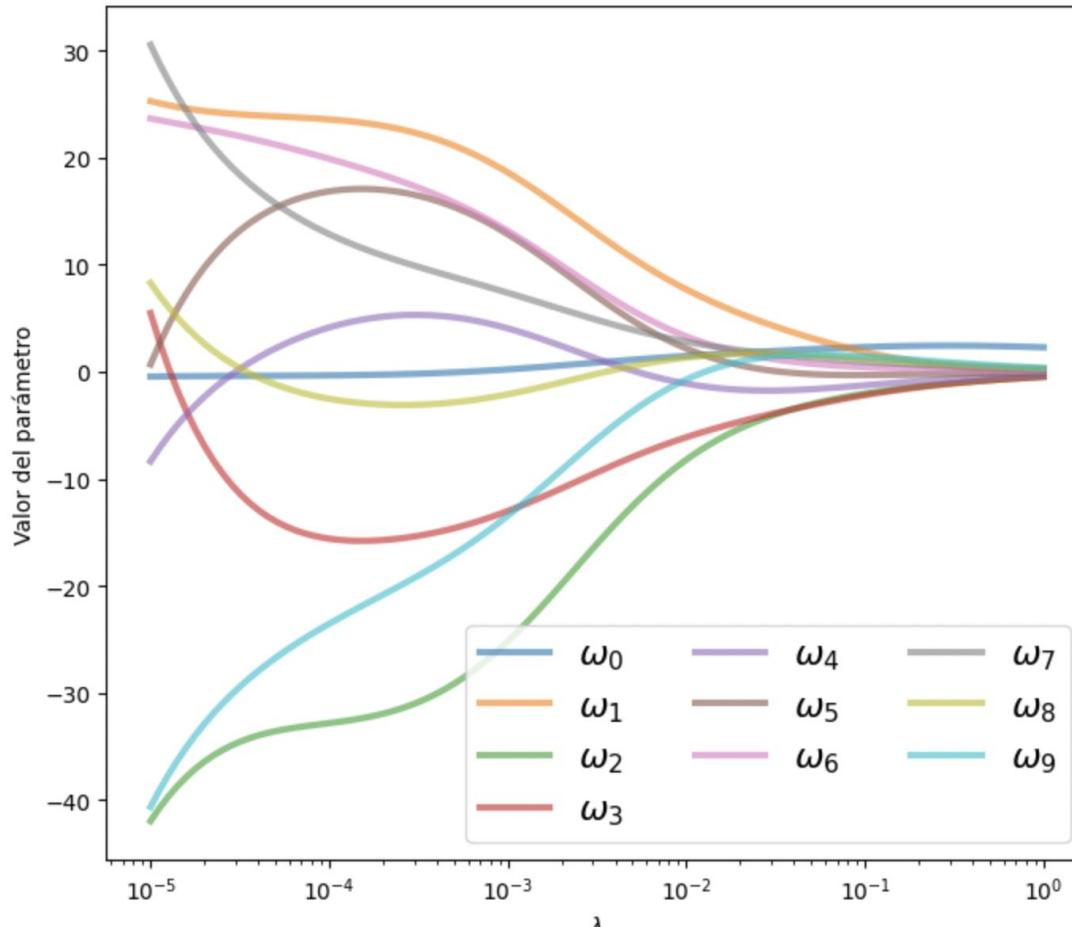
$$\|\boldsymbol{\omega}\|^2 = \boldsymbol{\omega}^T \boldsymbol{\omega} = (\omega_1 \dots \omega_M) \begin{pmatrix} \omega_1 \\ \vdots \\ \omega_M \end{pmatrix} = \sum_{i=1}^M \omega_i^2 .$$

El parámetro de regularización  $\lambda$  constituye un *nuevo hiperparámetro del modelo*.

Grado: 9;  $\lambda$ : 1.00e-03



# Coeficientes



# Regularización LASSO

Otra regresión regularizada que se utiliza a menudo es la regresión **LASSO** (*least absolute shrinkage and selection operator / operador de reducción y selección mínima absoluta*), que selecciona de forma natural las variables más relevantes y produce modelos más parsimoniosos.

En lugar de penalizar la función de error utilizando la suma de los cuadrados de los parámetros del modelo, como en el caso anterior, **LASSO** explota la norma  $l1$ , que es simplemente la suma de los valores *absolutos* de los parámetros del modelo.

En otras palabras, la norma  $l1$  de un vector es, simplemente:

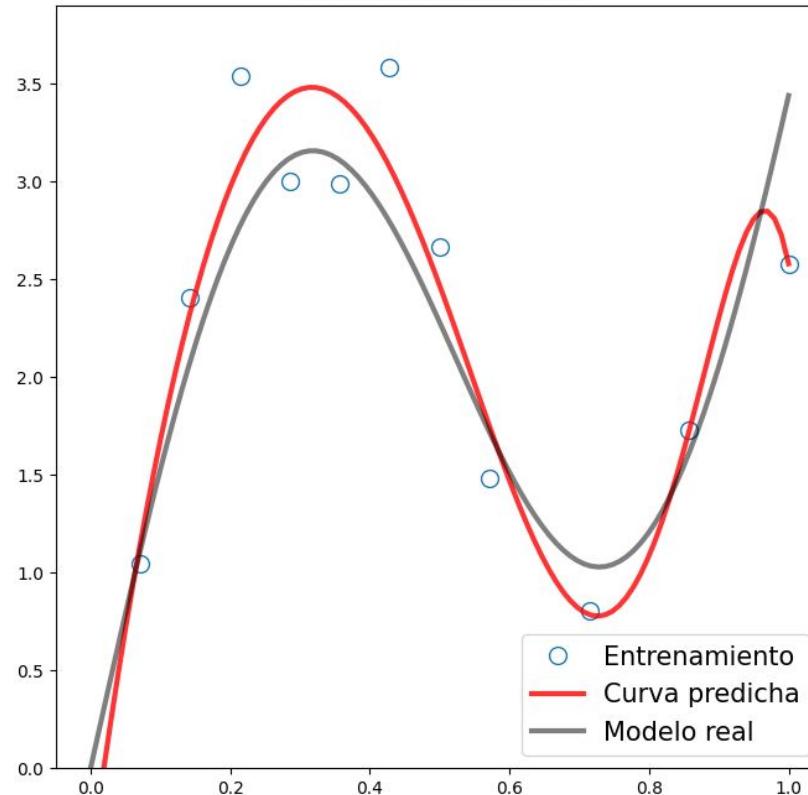
$$||\boldsymbol{\omega}||_1 = \sum_i |\omega_i| .$$

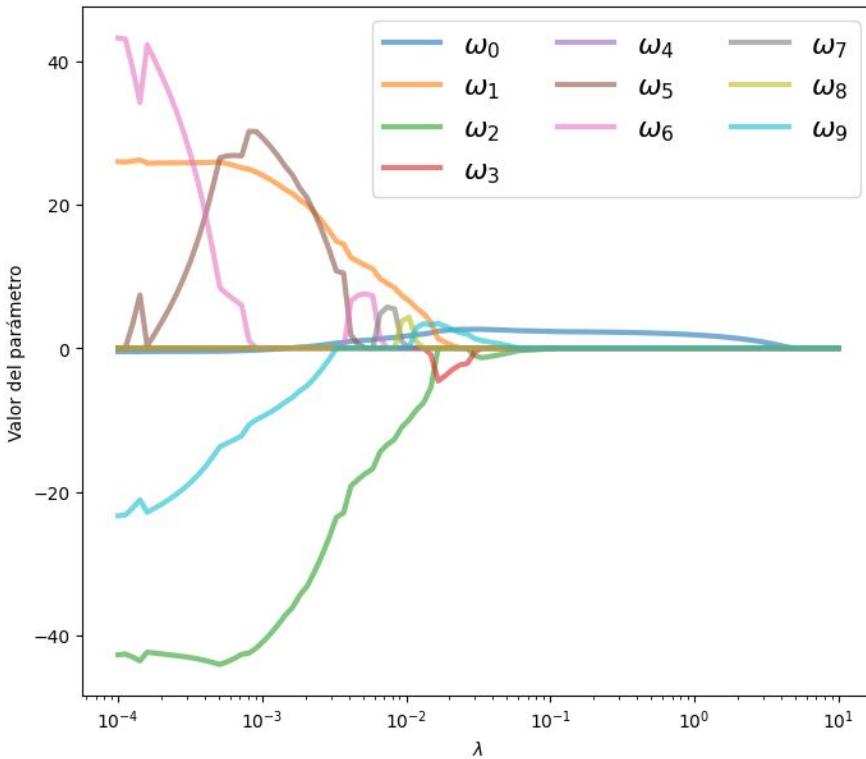
La función de error modificada es, por lo tanto,

$$E_{\text{lasso}}(\boldsymbol{\omega}; \lambda) = \frac{1}{2} \sum_{i=1}^N \{y(x_i, \boldsymbol{\omega}) - t_i\}^2 + \frac{\lambda}{2} \sum_{i=1}^M |\omega_i| ,$$

donde nuevamente introducimos el hiperparámetro  $\lambda$  para controlar el nivel de penalización.

Grado: 9;  $\lambda$ : 1.00e-05





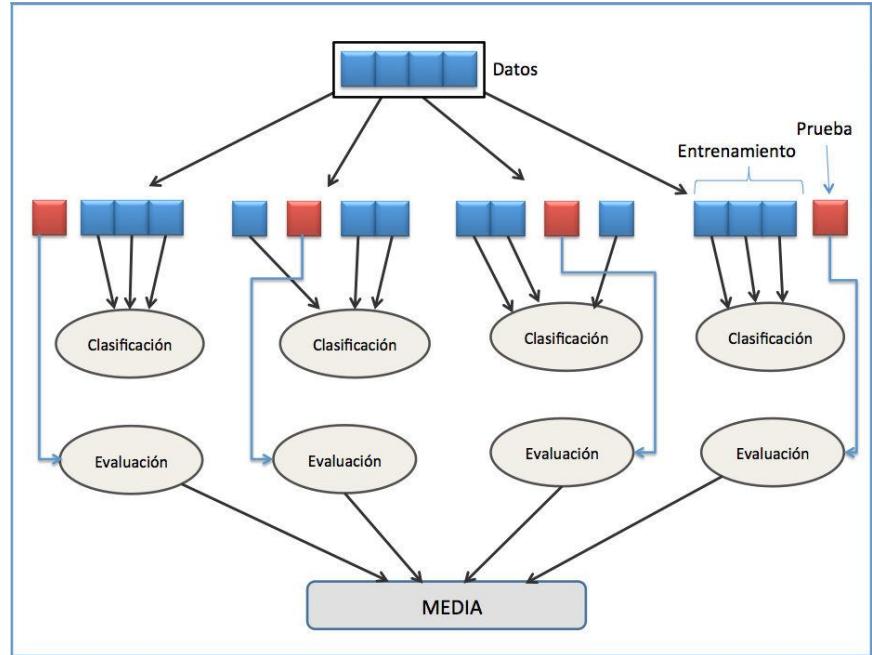
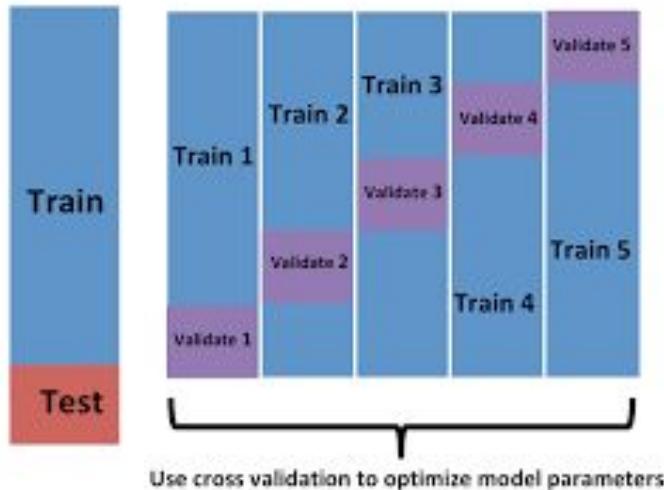
# Conjunto de validación

- ¿Cómo probar los resultados si usamos el conjunto de testeo para obtener los hiperparámetros?
- Conjunto de Entrenamiento/Train: donde ajustamos el modelo.
- Conjunto de Validación: Donde evaluamos el modelo y lo usamos para escoger mejores hiperparámetros/características etc.
- Conjunto de Prueba/Test: Lo usamos solamente al final del desarrollo para establecer la performance esperada del modelo en datos nuevos.



## Validación cruzada

Esquema k-fold cross validation, con k=4 y un solo clasificador.



- Fuente: Jean-Philippe Lang, [Predictors tutorial Archivado](#) el 3 de enero de 2014 en [Wayback Machine](#)., Bioinformatic Department Projects

# Grid search CV + k-fold cross validation en pocas líneas...

```
[ ] from sklearn.model_selection import GridSearchCV  
  
from sklearn.pipeline import Pipeline  
from sklearn.preprocessing import PolynomialFeatures,StandardScaler  
from sklearn.linear_model import LinearRegression, Ridge  
  
model = Pipeline([('scaler', StandardScaler()),  
                  ('polynomial', PolynomialFeatures()),  
                  ('regressor', Ridge(fit_intercept=False))  
                 ])  
  
# Definimos los parámetros para cada parte del pipeline. Separamos parámetro de nombre con '_':  
parameters = {'polynomial_degree' : range(1, 10),  
              'regressor_alpha': np.logspace(-2, 4, 100)}  
  
grid_search = GridSearchCV(model, parameters, scoring='neg_mean_squared_error', cv=5, n_jobs=-1)
```

Ahora corremos `fit` para saber cuáles son los mejores hiperparámetros.

```
[ ] grid_search.fit(x_train_full, t_train_full)
```

Los mejores hiperparámetros son:

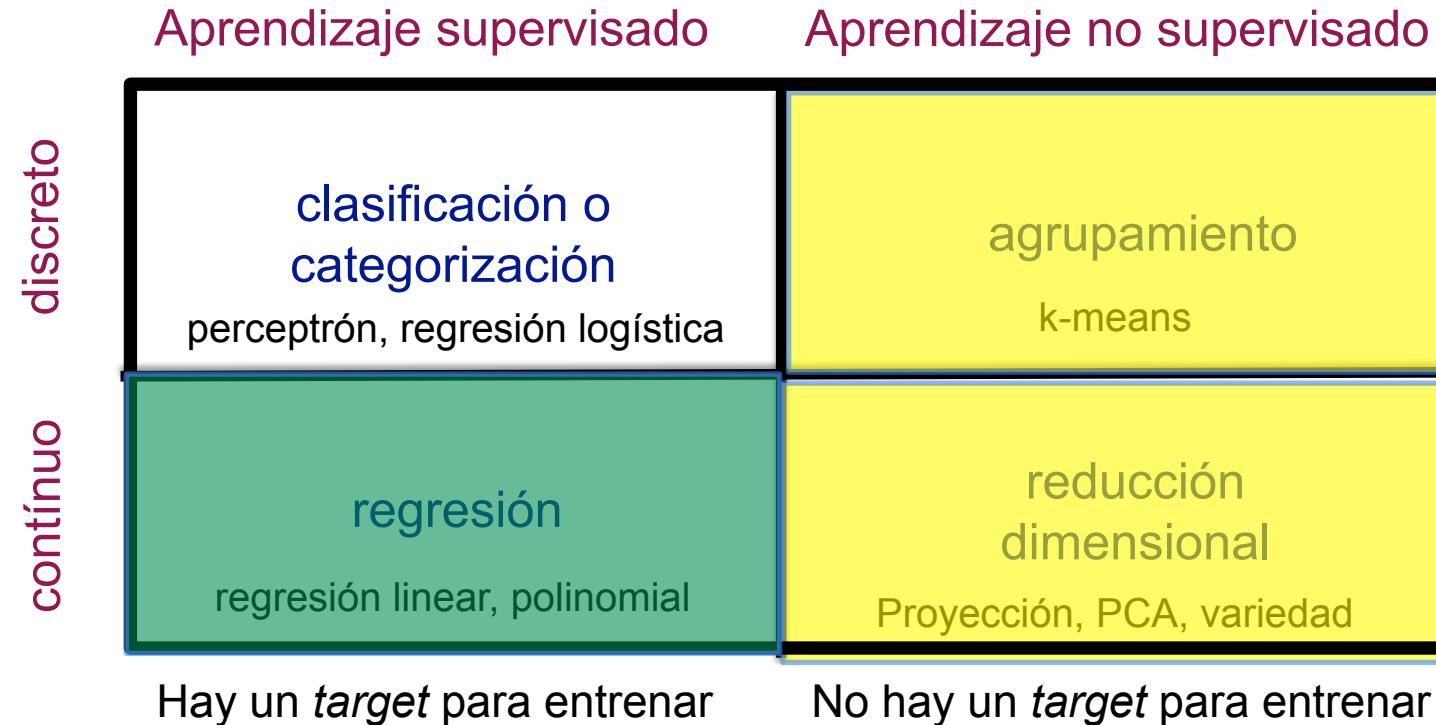
```
[ ] grid_search.best_params_  
{'polynomial_degree': 2, 'regressor_alpha': 24.770763559917114}
```



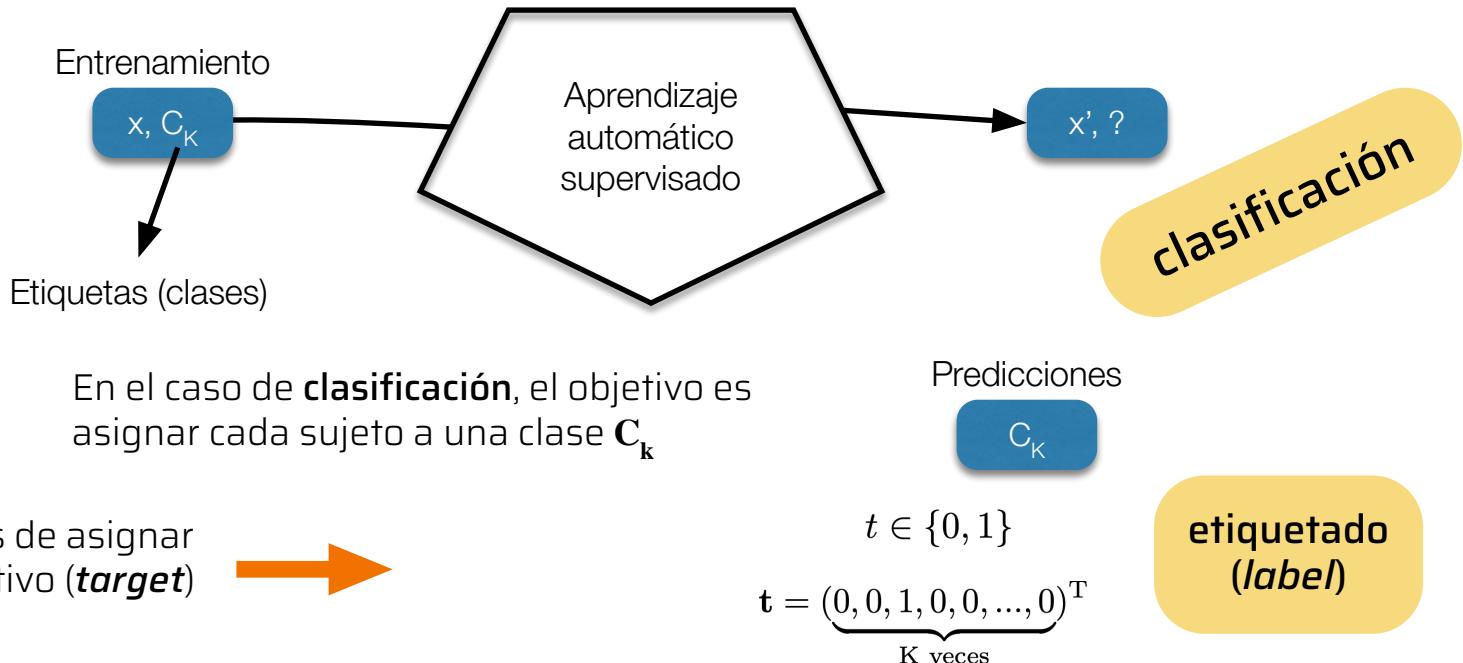
Argentina  
programa  
4.0

# Clasificación

# Tipos de datos y problemas

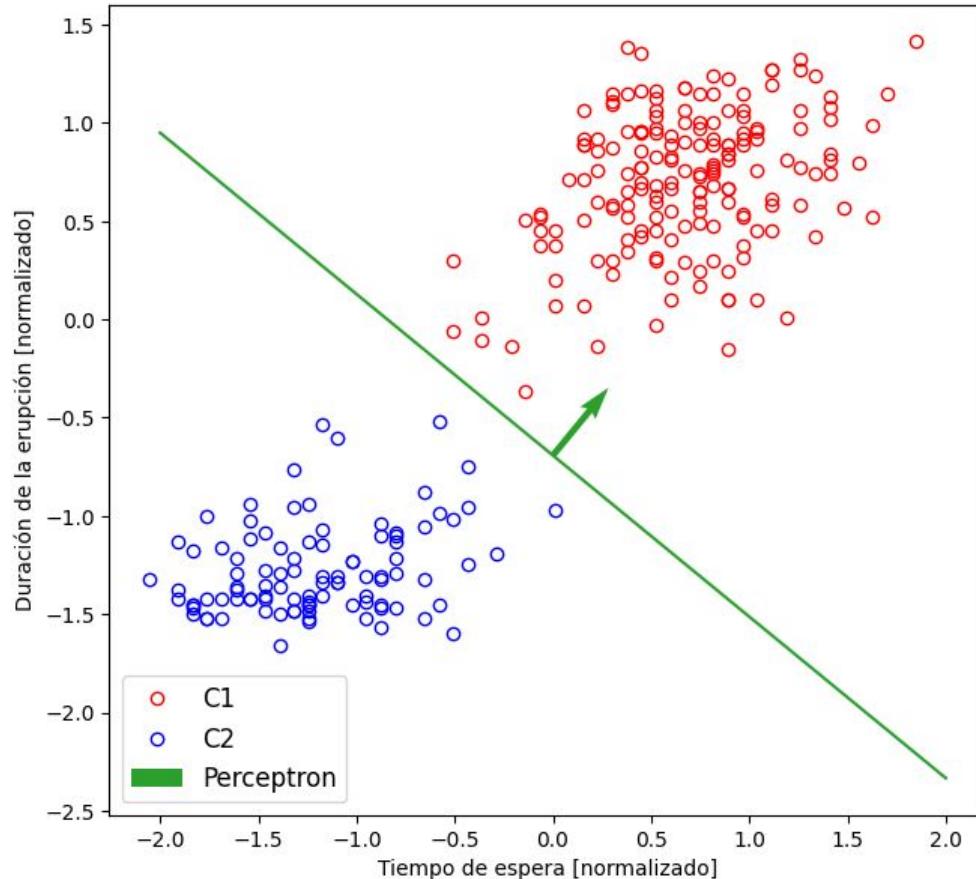


# Aprendizaje supervisado: regresión x clasificación



# Clasificación de un conjunto de datos en 2 dimensiones

Atención:  
ahora el target  
es 0 o 1



# Fronteras de decisión y regiones

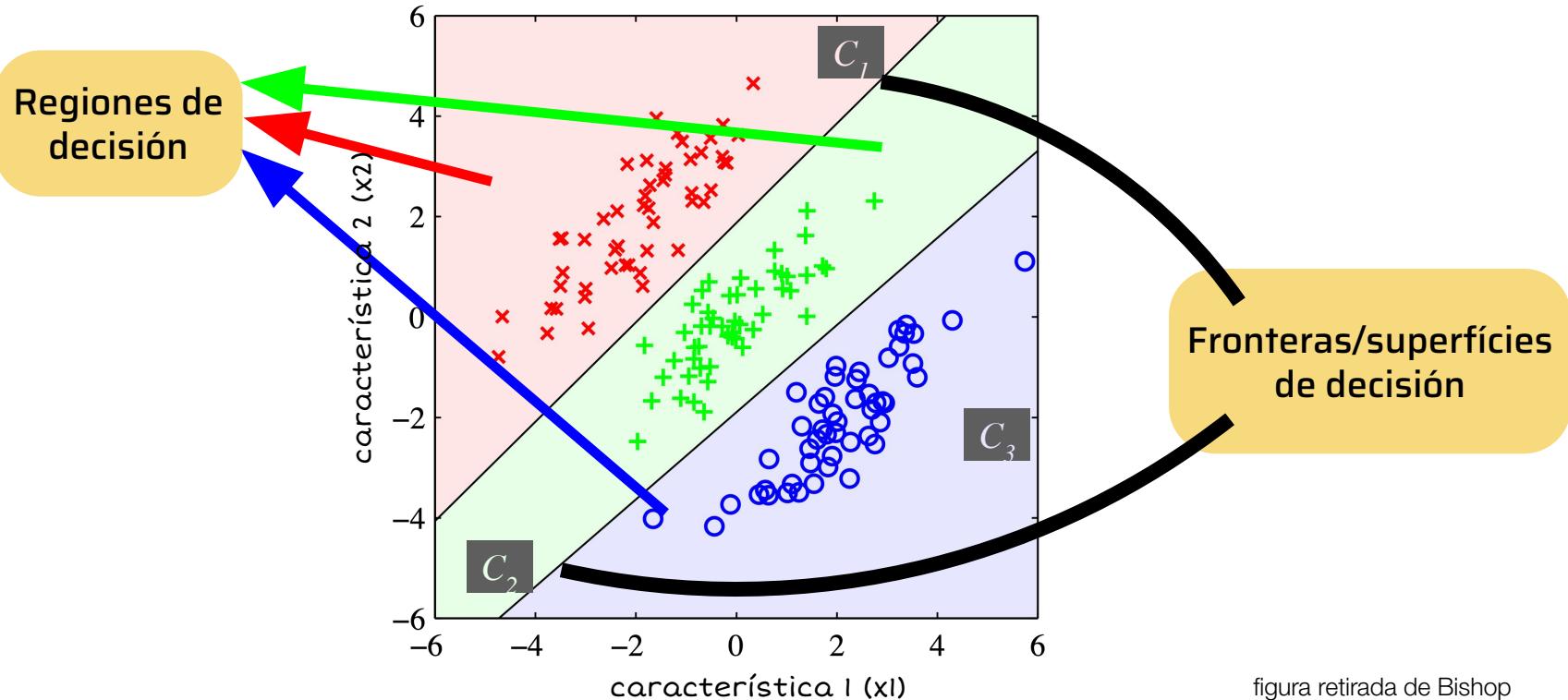
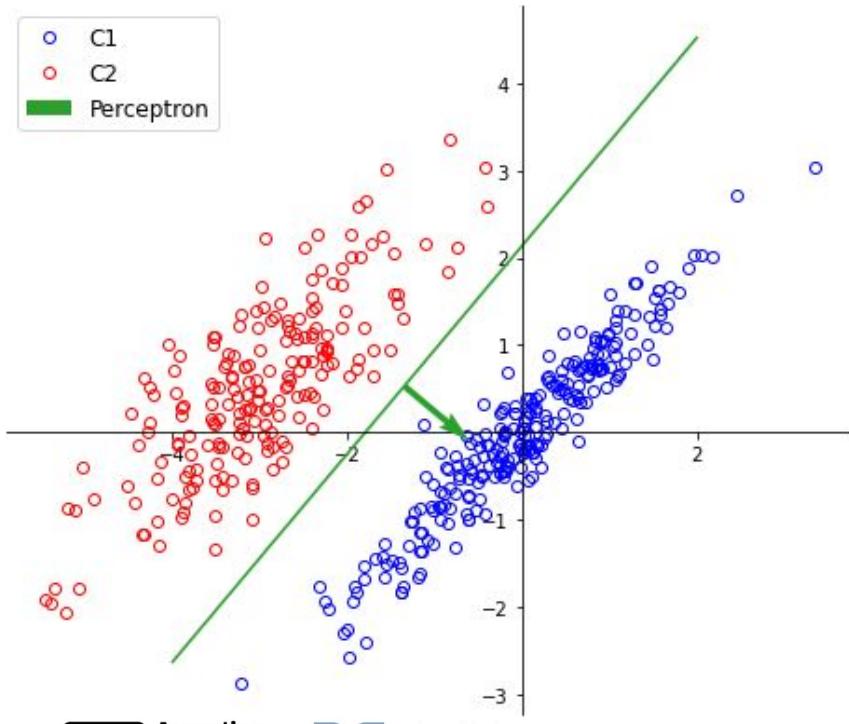
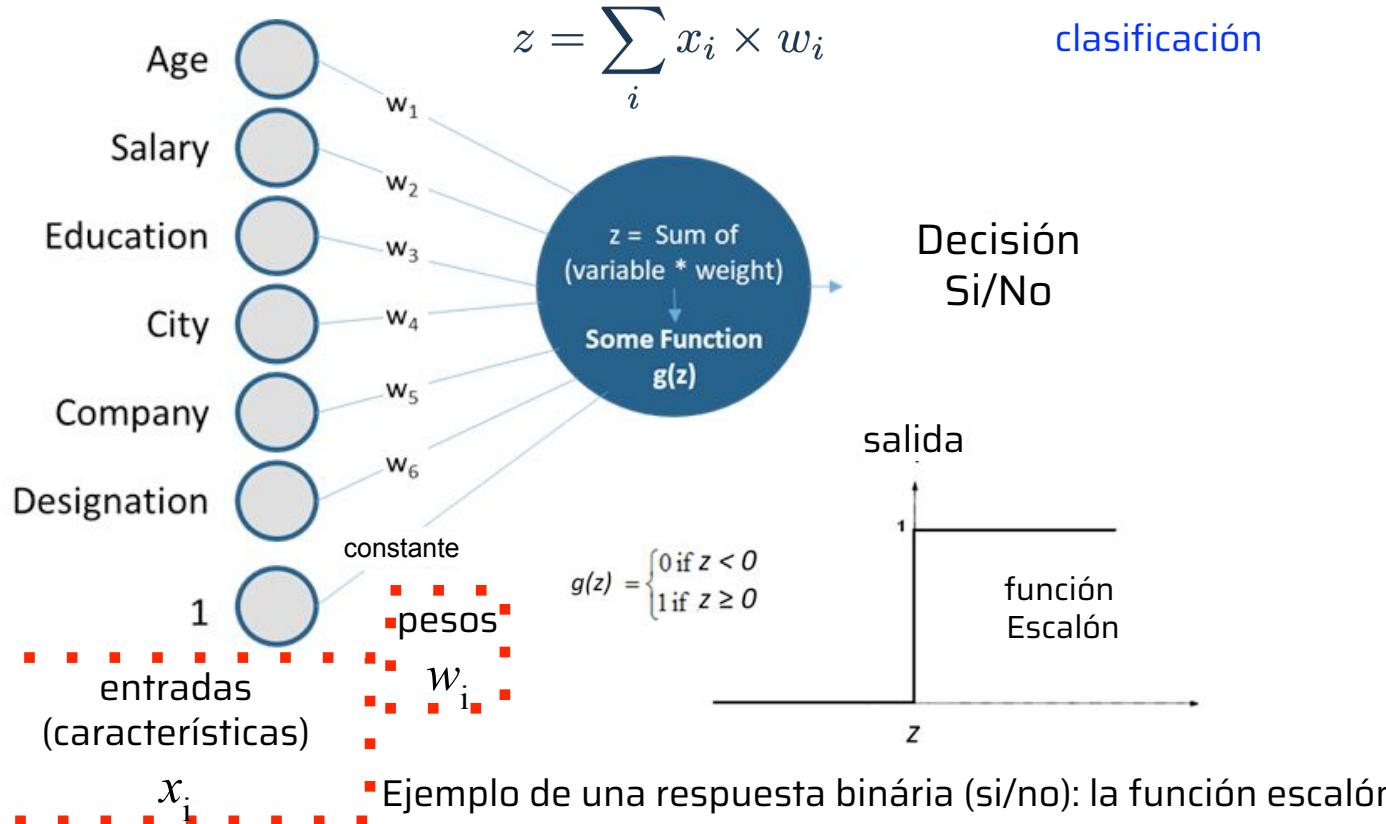


figura retirada de Bishop

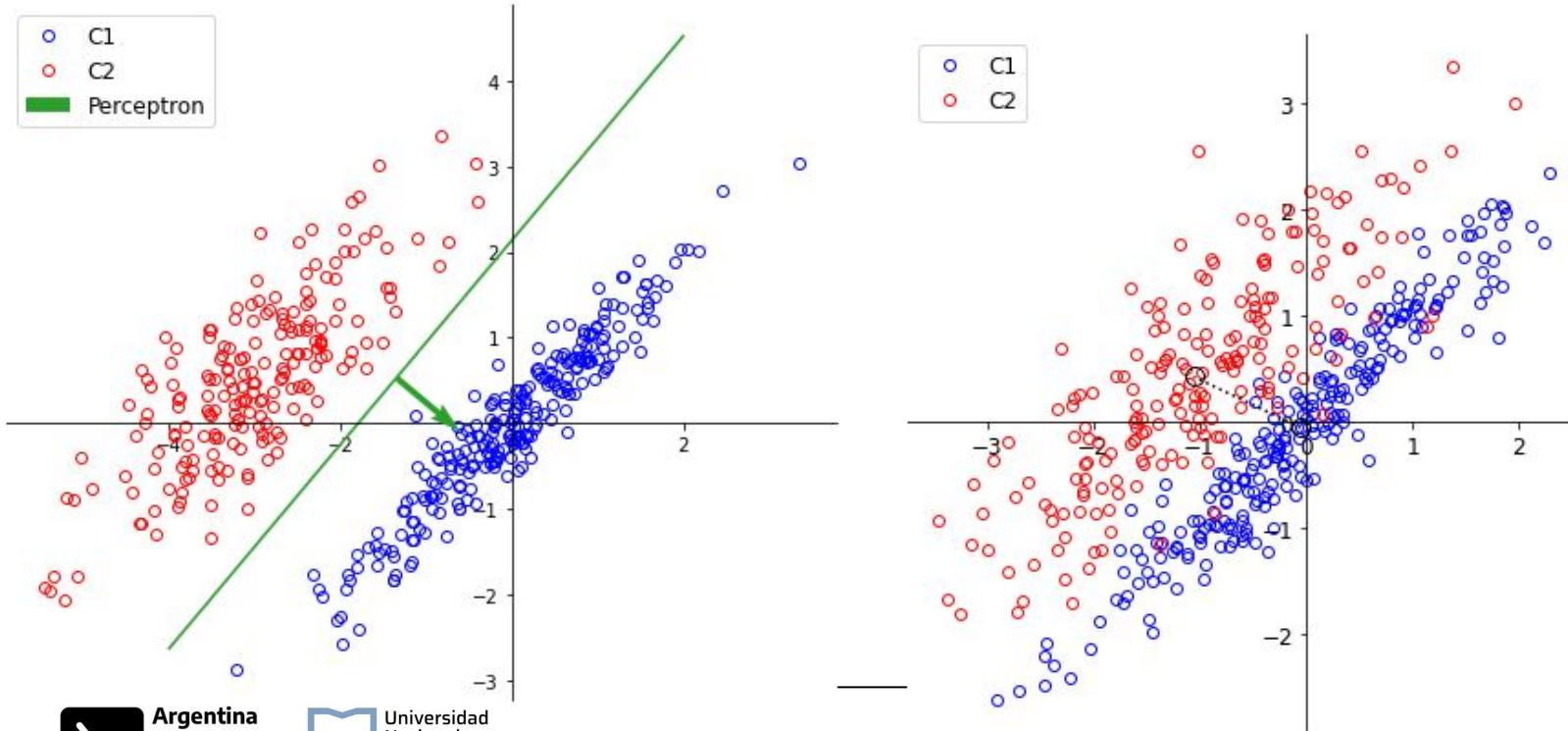
# Conjuntos de datos linealmente separables



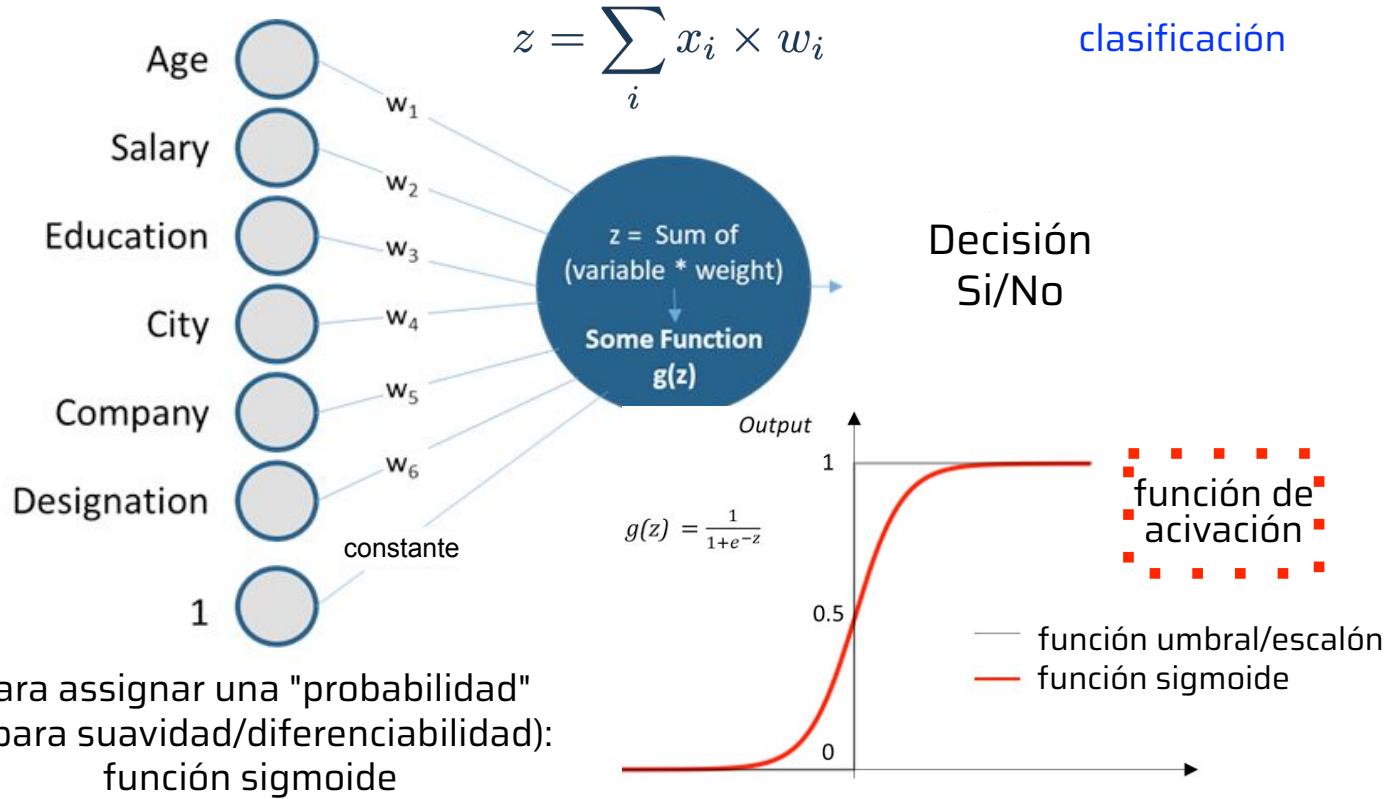
# El perceptrón



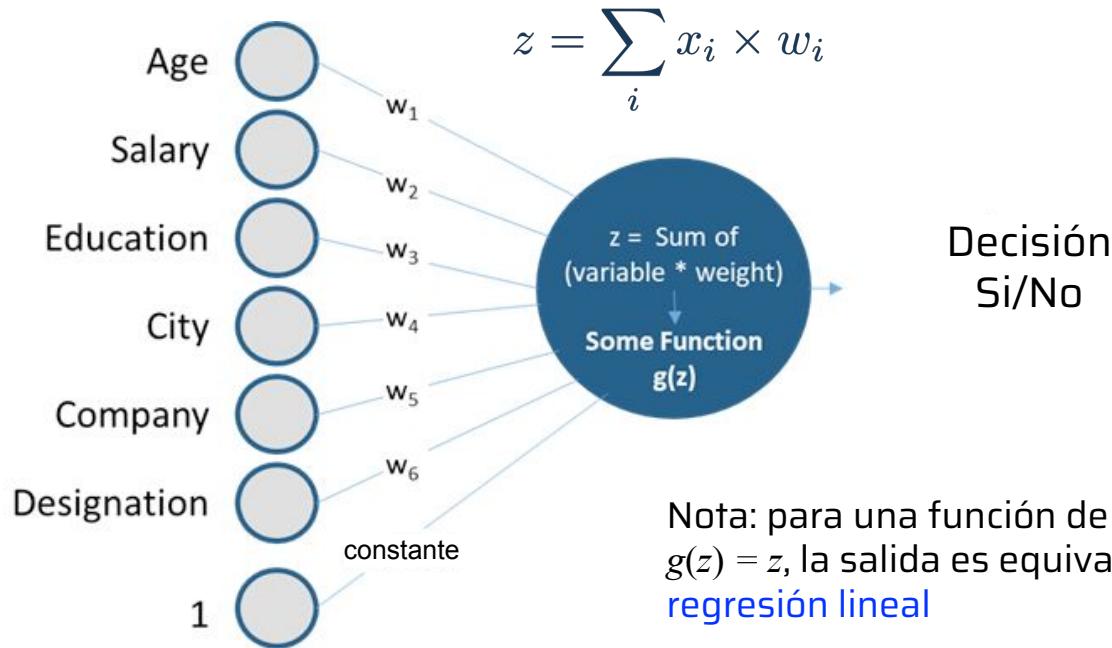
# Conjuntos de datos linealmente separables



# Regresión logística

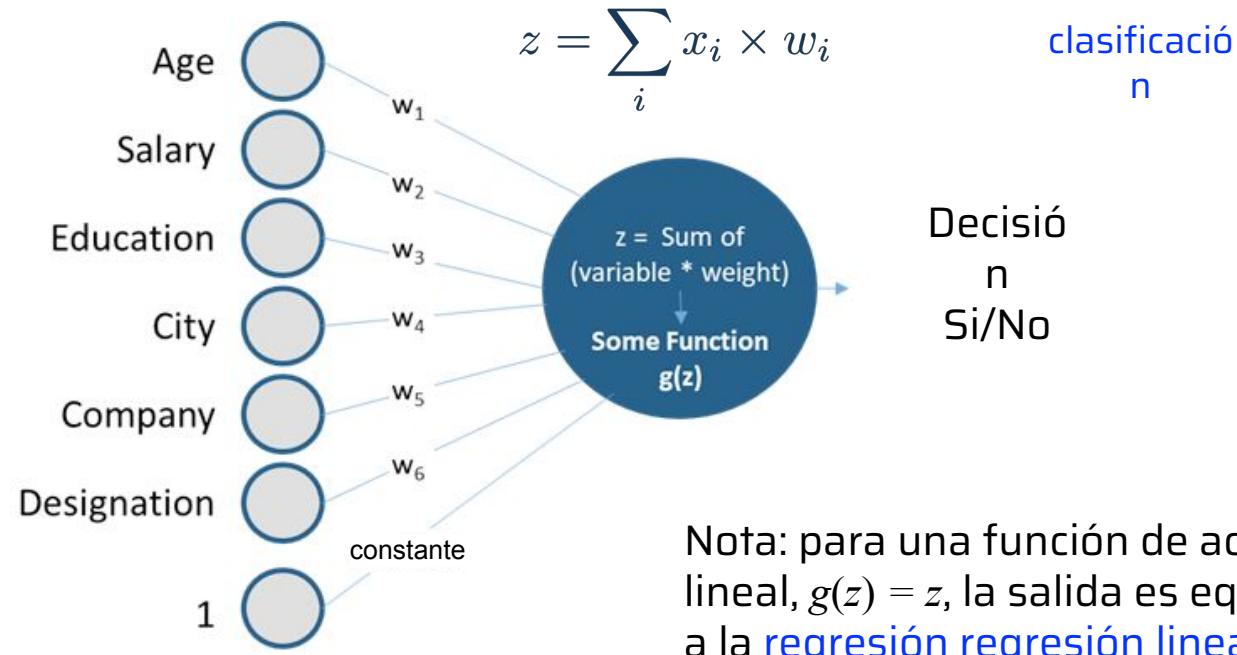


# Regresión lineal

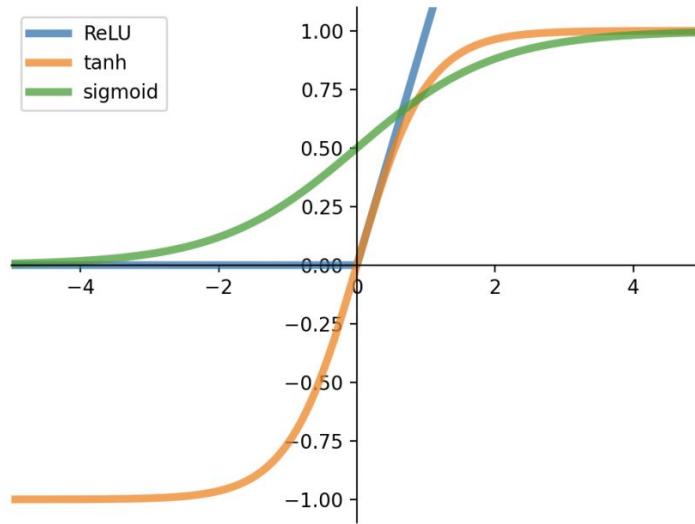


Nota: para una función de activación lineal,  
 $g(z) = z$ , la salida es equivalente a la  
**regresión lineal**

# El perceptrón



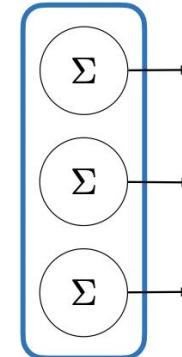
# Observación: tipos de función de activación



Capas internas: ReLU (o SeLU, GeLU)  
(problema de los gradientes evanescentes)

Capa de salida			
Problema	Tamaño	Activación	Error
Regresión	$N$	$f(x) = x$	MSE RMSE
Clasificación Binaria	$I$	$f(x) = \text{sigmoide}$	Cross-entropy
Clasificación Multi-class	$K$	$f(x) = \text{softmax}$	Multiclass Cross-entropy

$$s(z_i) = \frac{\exp(z_i)}{\sum_{k=1}^K \exp(z_j)}$$



# ¿Cómo evaluar los resultados de clasificación?

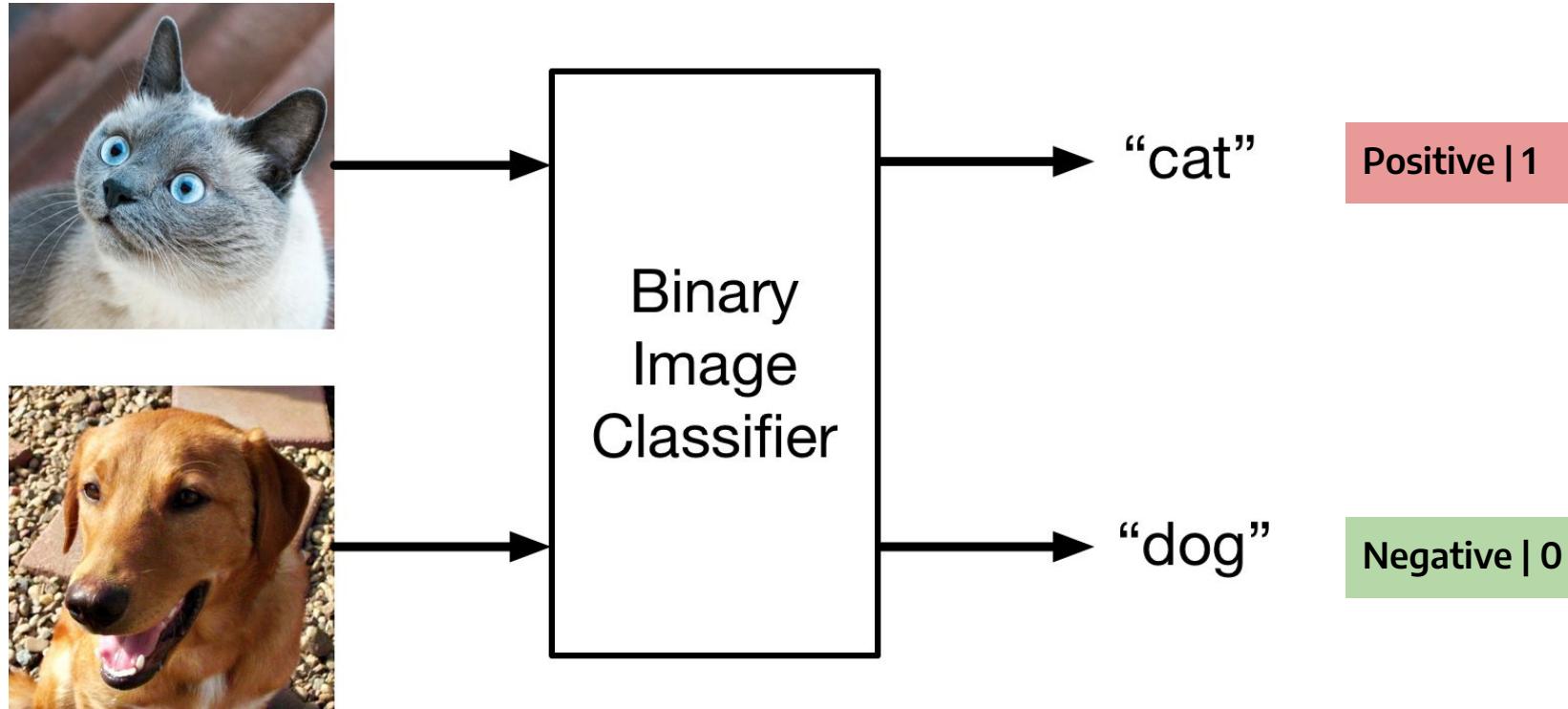
En la regresión el mismo MSE nos ofrecía una forma de evaluar los resultados

En clasificación la comparación es con 0 o 1 y se usan otras métricas para evaluar los resultados.

La respuesta es binaria, pero la evaluación de los resultados no se hace con un solo número!

Además, está el umbral de decisión, un nuevo *hiperparámetro*!

# Un modelo de clasificación fácil



# Métricas de clasificación

Matriz de confusión

$$\begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$$

elementos relevantes

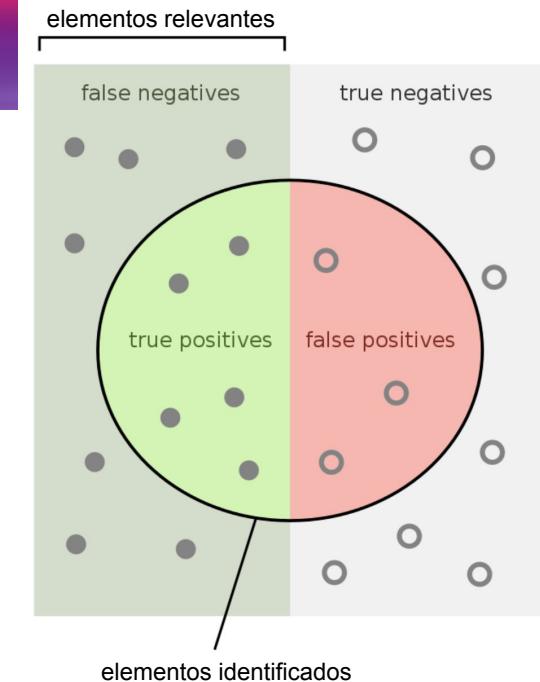
Métricas

$$\text{precision} = \frac{TP}{TP + FP} = \frac{\text{}}{\text{} + \text{}}$$

=

$$\text{recall} = \frac{TP}{TP + FN} = \frac{\text{}}{\text{} + \text{}}$$

=



Tasa de éxitos

$$\text{accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

## Qué queremos maximizar?

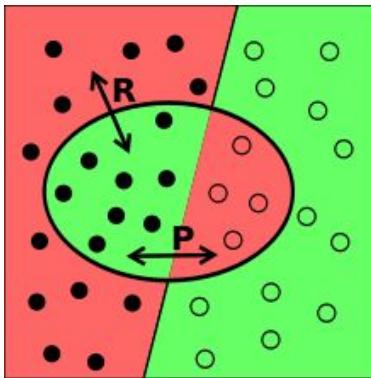
Precision: Fracción de emails importantes entre los emails *que llegaron al Inbox*.

Precision = Positive Predicted Value  
PPV = TP / (TP + FP)

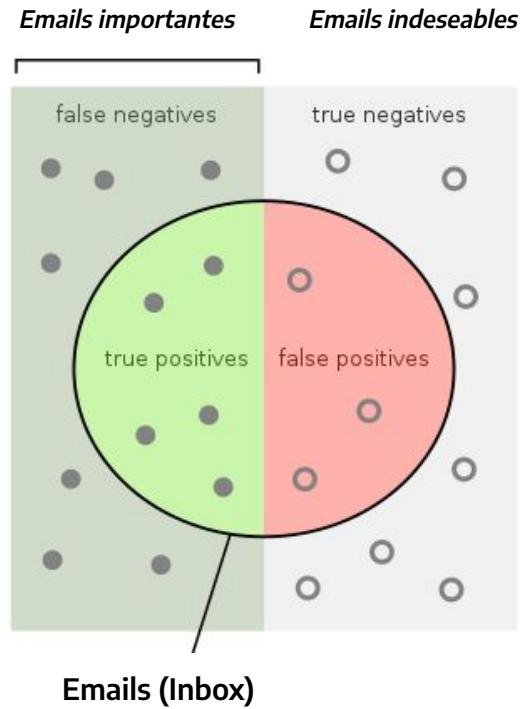
Exhaustividad (Recall): Fracción de *emails importantes que llegaron al inbox*, con respecto a **todos los emails importantes**.

Recall = True Positive Rate  
TPR = TP / P = TP / (TP + FN)

Accuracy = Exactitud  
ACC = TP + TN / P + N



verde = acierto  
rojo = error



$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$
$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

## Qué pasa si dejamos entrar todos los emails al Inbox?

Precision: Fracción de **emails importantes** entre los emails **que llegaron al Inbox**.

Precision = Positive Predicted Value  
 $PPV = TP / (TP + FP) = 12 / 22 = 54\%$



Exhaustividad (Recall): Fracción de **emails importantes que llegaron al inbox**, con respecto a **todos los emails importantes**.

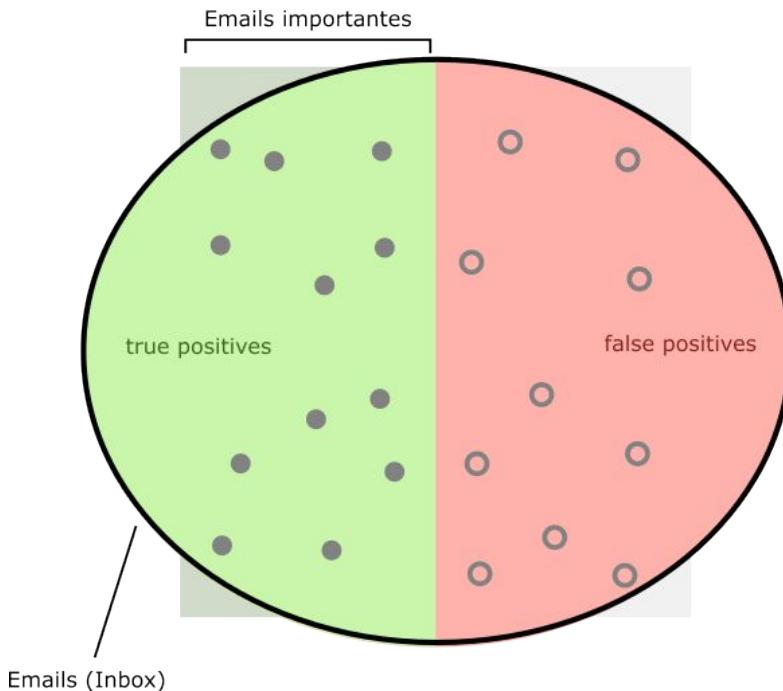
Recall = True Positive Rate  
 $TPR = TP / P = TP / (TP + FN) = 12 / 12$

~~100~~ !!

Pero también:

False Positive Rate (Probabilidad de falsa Alarma)  
 $FPR = FP / N = FP / (FP + TN) = 10/10$

~~100~~ !!



Cuántos emails en el Inbox son importantes?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Cuántos emails importantes hay en el Inbox?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Cuántos emails no deseables hay en el Inbox?

$$\text{Falsa Alarma} = \frac{\text{false positives}}{\text{false positives} + \text{true negatives}}$$

## Qué pasa si dejamos entrar pocos emails al Inbox?

### Precision (Especificidad)

Fracción de **emails importantes** entre los emails **que llegaron al Inbox**.

Precision = Positive Predicted Value  
PPV = TP / (TP + FP) = 1 / 1 = 100%

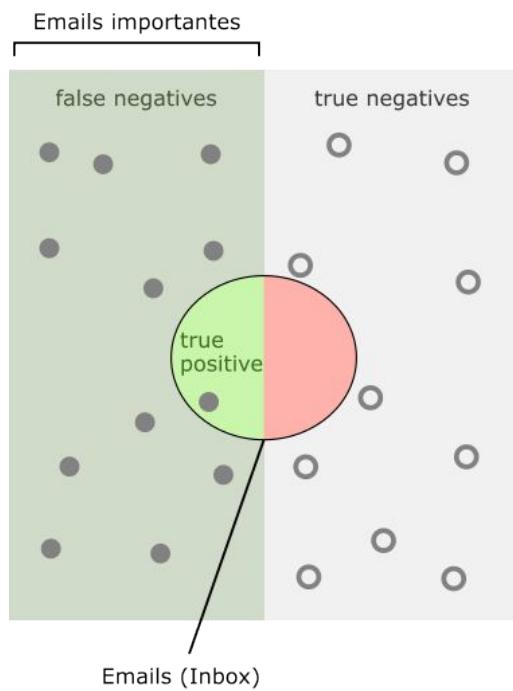
~~100~~ !!

### Exhaustividad (Recall)

Fracción de **emails importantes que llegaron al inbox**, con respecto a **todos los emails importantes**.

Recall = True Positive Rate

TPR = TP / P = TP / (TP + FN) = 1 / 12 = 8.33% 😞 !!



Cuántos emails en el Inbox son importantes?

$$\text{Precision} = \frac{\text{green}}{\text{green} + \text{red}}$$

Cuántos emails importantes hay en el Inbox?

$$\text{Recall} = \frac{\text{green}}{\text{green} + \text{grey}}$$

# De donde viene la matriz de confusión?

La “**matriz de confusión**” se usa para **evaluar algoritmos** de clasificación/predicción, y ver en qué casos **confunde** clases. He ahí la confusión. Si hay 6 perros reales y el algoritmo predice o clasifica bien solamente 3, se está **confundiendo** en otros 3 casos.

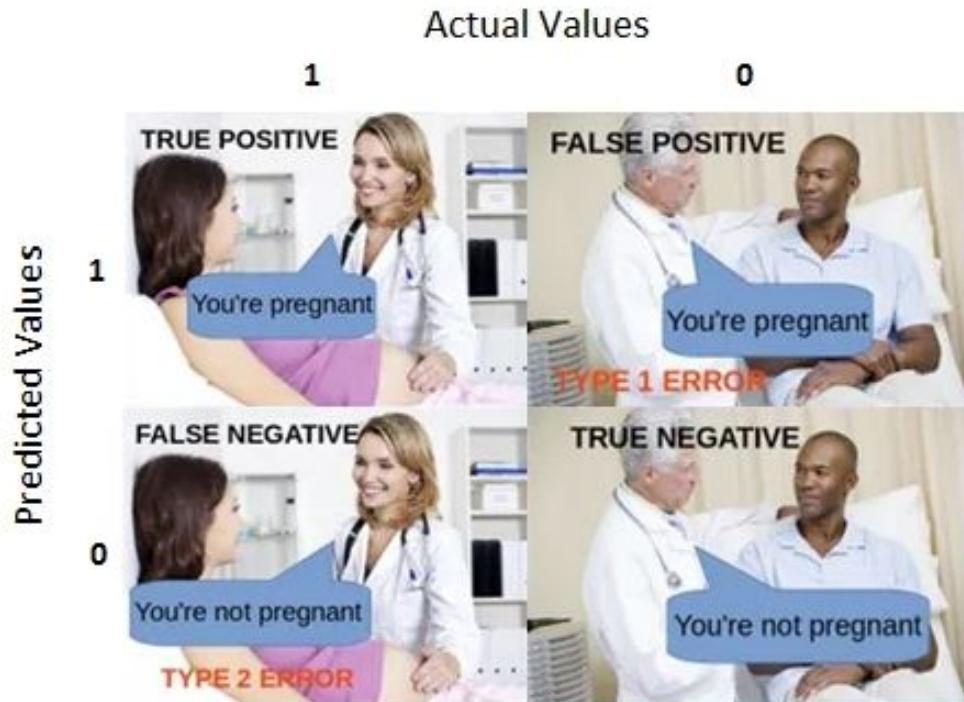
Se la llama también **matriz de error**, aunque también muestra los **aciertos!** Con total validez se la podría llamar "Matriz de Equivocaciones y Aciertos del algoritmo".

*Aunque sería un poco largo, demasiado claro y poco perverso.*

# Caso fácil

PREDICTED VALUES

ACTUAL VALUES	
TP	FP
FN	TN

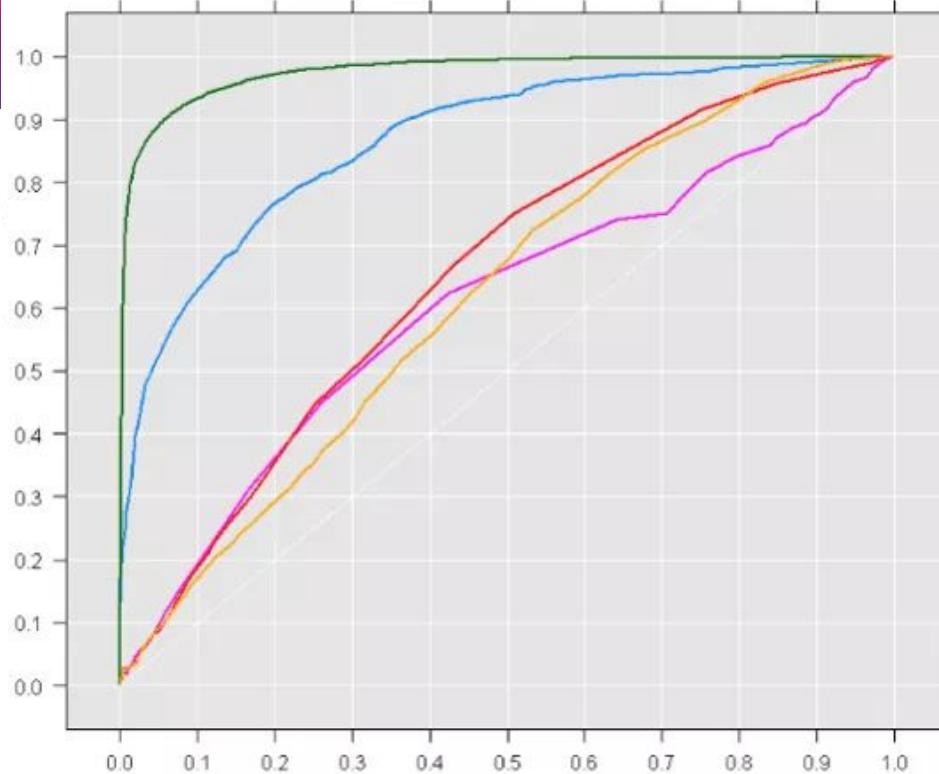


# Caso difícil: cancer, pacientes (vida, muerte)

Predicción de Pronóstico de progresión de Cancer.

ESTADO REAL		PRONOSTICO	
<i>Benigno</i>	<i>Maligno</i>		
<i>Benigno</i>	TP	FP	
<i>Maligno</i>	FN	TN	

ESTADO REAL		PRONOSTICO	
<i>Benigno</i>	<i>Maligno</i>		
<i>Benigno</i>	26	3	
<i>Maligno</i>	0	57	



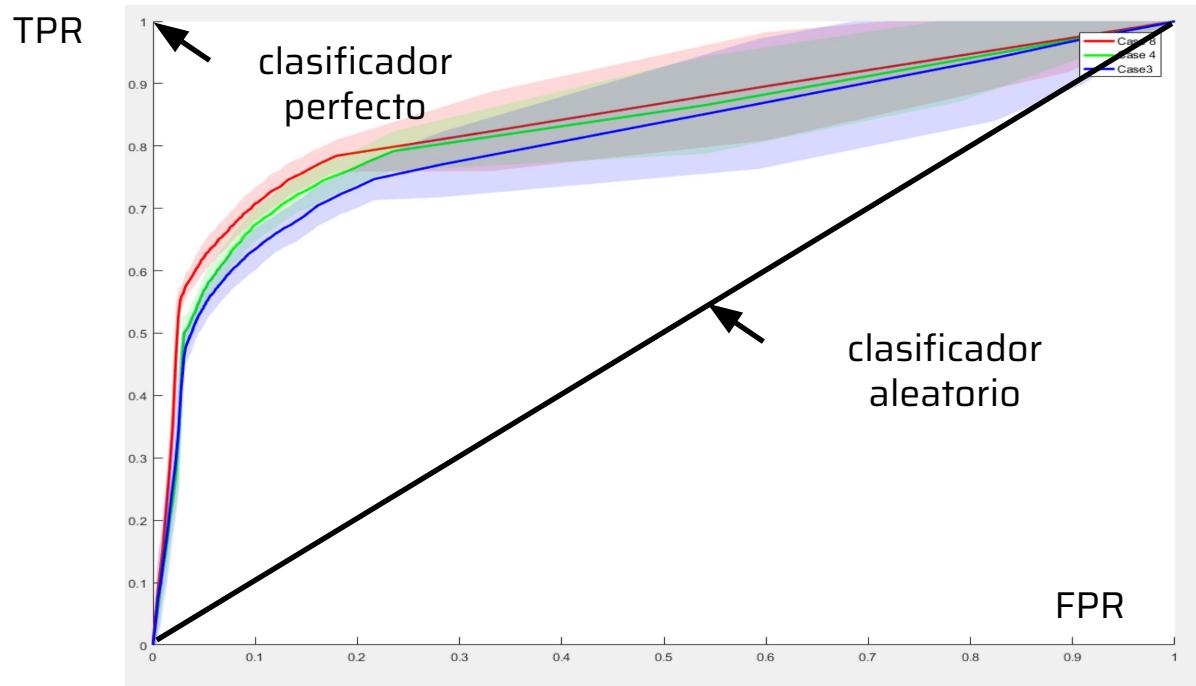
exhaustividad : Sensitivity (recall, true positive rate) refers to the probability of a positive test, conditioned on truly being positive.

### fall-out or false positive rate (FPR) - 1 specificity

- Specificity (true negative rate) refers to the probability of a negative test, conditioned on truly being negative.

# La curva Receptor - Operador (Receiver Operator - ROC curve)

- TPR x FPR em función del *working point*



Área bajo la curva  
(area under the curve - AUC)

Comparación entre  
métodos

# Regresión x Clasificación

- **Principal diferencia:**
  - Objetivo: separar en clases distintas x predecir un número
  - (Reducir a separar entre dos clases o predecir un número)
  - Datos de entrenamiento: 0 o 1 (u otros 2 valores) x números continuos
  - ¡Lo que cambia es la función de pérdida!
- **Conceptos "universales": fit, hiperparámetros, regularización, función activación, etc.**
- **Métodos: grid search, CV**
- **Métricas**

# Módulo 2

Hemos aprendido conceptos y prácticas de ciencia de datos

Por sí solo son muy poderosos y nos abren las puertas al mundo de los datos, a explorar, comprender, predecir, etc.

Son también pilares del aprendizaje automático, del aprendizaje profundo y de los métodos tan en voga de "inteligencia artificial"



**Argentina  
programa  
4.0**

# PLAN DE ESTUDIOS

Introducción

## Programación en Python

Dictado del 13/02 al 19/04. **Lunes y  
Miércoles de 18 a 20 hs.**

[Ver mas](#)

Intermedio

## Ciencia de Datos

Dictado del 8/05 al 12/07. **Lunes y  
Miércoles de 18 a 20 hs.**

[Ver mas](#)

Especialización

## Aprendizaje Automático

Dictado del 7/08 al 16/10. **Lunes y  
Miércoles de 18 a 20 hs.**

[Ver mas](#)



Universidad  
Nacional  
de San Martín



Escuela de  
Ciencia y Tecnología  
ECyT\_UNSAM



Secretaría de Economía  
del Conocimiento