

Introducción a la Inteligencia Artificial
Facultad de Ingeniería
Universidad de Buenos Aires



Índice

1. Terminology
2. Pipeline
3. Train-test-validation
4. Feature engineering
5. Regresión lineal



Input Analysis - Machine Learning Pipelines

Machine Learning Terminology

- Raw vs. Tidy Data $\xrightarrow{\text{raw data: datos crudos}}$ $\xrightarrow{\text{Tidy data: data preprocesada (limpieza, agrupación, transf...)}}$
- Training vs. Holdout Sets \rightarrow divisiones del dato original que utilizamos para 'aprender'
- Baseline \rightarrow ①
- Parameters vs. Hyperparameters
- Classification vs. Regression $\rightarrow f_{y|x}(y|x) \vee f_x(x)$
- Model-Based vs. Instance-Based Learning
- Shallow vs. Deep Learning \rightarrow ②

① baseline: es un conjunto de modelos super basicos que nos indican un limite

(ej de modelos baseline: media, regresión lineal, CART, RF (regressor))

baseline interna

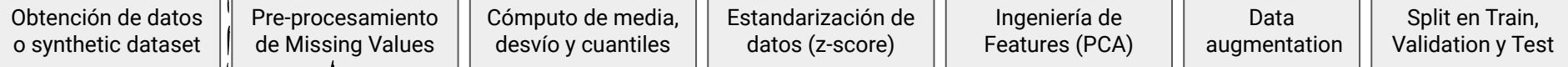
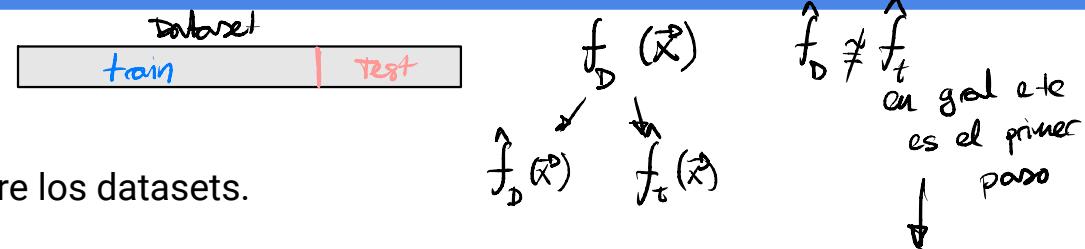
② \rightarrow shallow Learning \ddagger HP (~ 20 hiperparámetros)
Deep Learning $\ddagger\ddagger\ddagger\ddagger\ddagger$ HP ($\sim 1M$ hiperparámetros)



Dataset pipeline

Acciones que generalmente se ejecutan sobre los datasets.

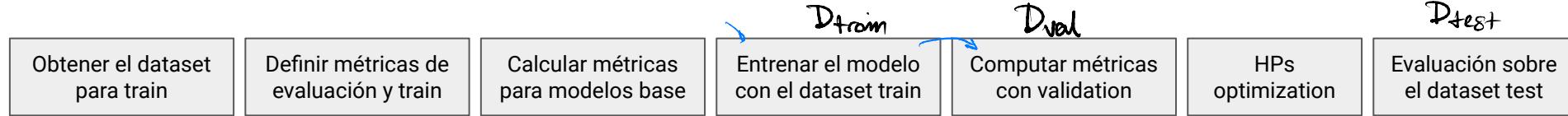
DE \leftarrow DS



son métodos que se basan en est. básica. → preprocesamientos → parámetros θ_n (training)

Model pipeline

Pasos involucrados al entrenar un modelo de Machine Learning



{ baseline: Random Forest (hiperparam. fijos) F_{1b}
models: [Knn, LDA, K SVM] \rightarrow [$m_2(0), m_2(0), m_3(0)$]
metrica: F1



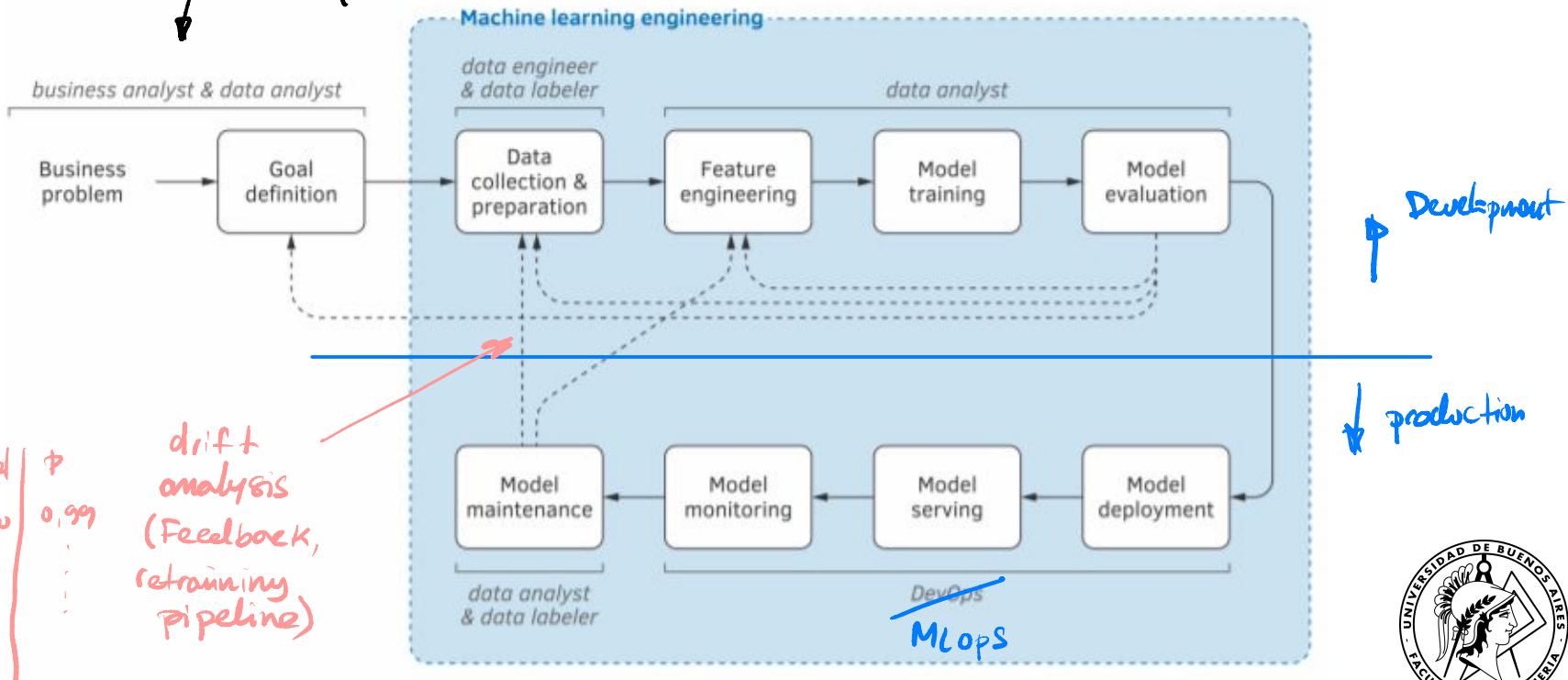
dataset de validación = dataset dev

Iteration	F_1_{train}	F_1_{dev}	HP	Rank
baseline	0,58	0,5	[--]	33
Knn 1	0,9	0,89	[--]	1
LDA 1	0,68	0,6	[--]	3
SVM 1	0,7	0,7	[--]	12
N				
N				
N				

Input Analysis - Machine Learning Pipelines

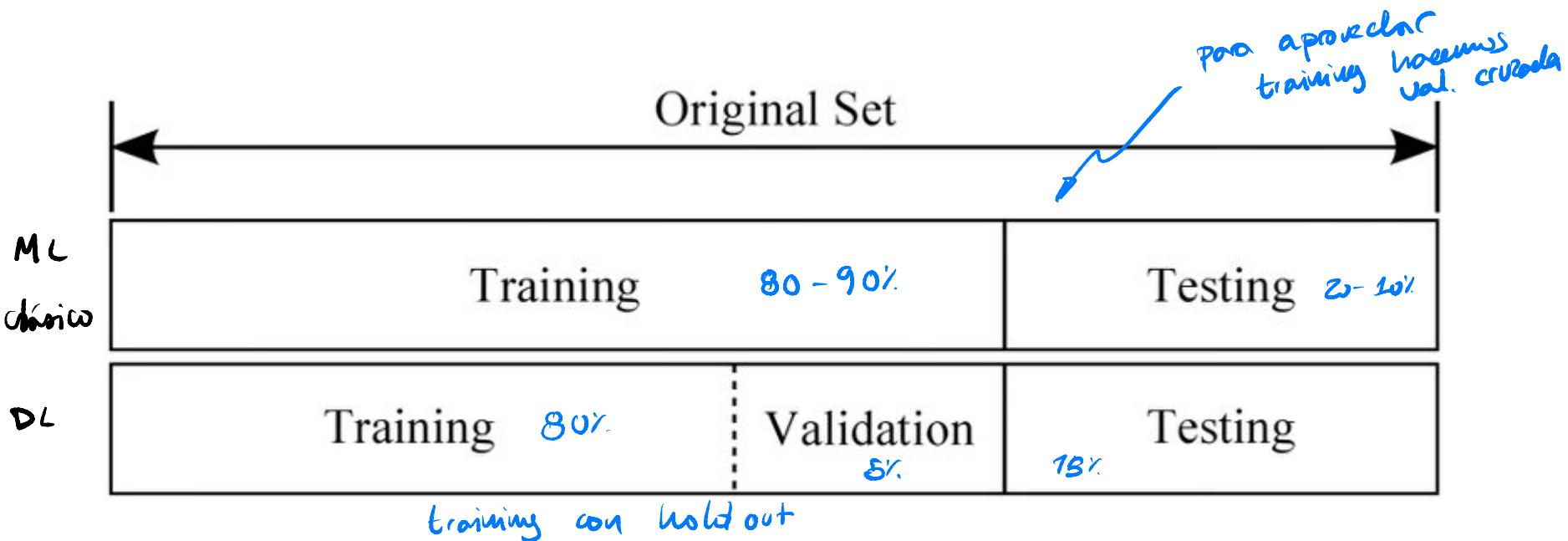
Machine Learning Pipeline

c que queremos
responder?



Ingeniería de Features

Train - test - validation



idealmente cada set es una muestra representativa de mis datos.

model (φ)

φ : conj. de HP

	Training				Test
	1	2	3	4	
it 1	val	tr	tr	tr	$\rightarrow m_{(1)} (F_1)$
it 2	tr	val	tr	tr	$\rightarrow m_{(2)}$
it 3	tr	tr	val	tr	$\rightarrow m_{(3)}$
i + 4	tr	tr	tr	val	$\rightarrow m_{(4)}$

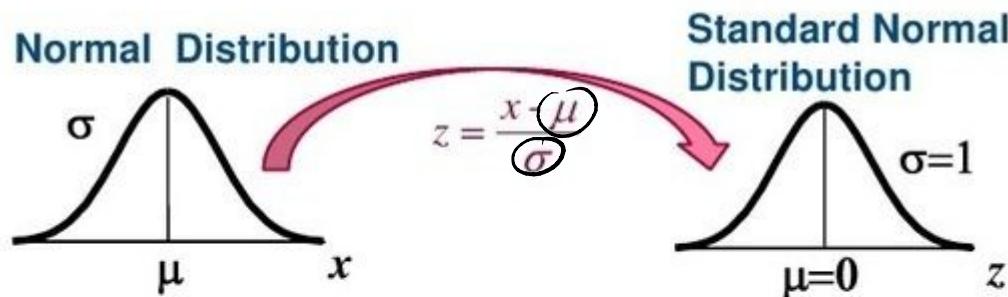
en cada iteración:

model (φ). fit (todo subset que sea train)

model predict (val it) \Rightarrow metrica $m_{(it)}$

Normalización

Muchos algoritmos de Machine Learning necesitan datos de entrada centrados y normalizados. Una normalización habitual es el z-score, que implica restarle la media y dividir por el desvío a cada feature de mi dataset.



$$f: \mathbb{R} \rightarrow \mathbb{R}$$
$$x \rightarrow z$$
$$\hat{z} = \frac{x - \bar{x}}{s^2} \quad ; \quad \bar{x}: \text{promedio}$$
$$s^2: \text{varianza muestral}$$

Ingeniería de Features - Missing Values

Missing Values

Es muy común en la práctica, recibir como datos de entrada, datasets que tienen información incompleta ("NaN").

ID	City	Degree	Age	Salary	Married ?
1	Lisbon	Nan	25	45,000	0
2	Berlin	Bachelor	25	Nan	1
3	Lisbon	Nan	30	Nan	1
4	Lisbon	Bachelor	30	Nan	1
5	Berlin	Bachelor	18	Nan	0
6	Lisbon	Bachelor	Nan	Nan	0
7	Berlin	Masters	30	Nan	1
8	Berlin	No Degree	Nan	Nan	0
9	Berlin	Masters	25	Nan	1
10	Madrid	Masters	25	Nan	1



Ingeniería de Features - Missing Values

Solución 1

Una forma de solucionar el problema es remover las filas y las columnas que contienen dichos valores.

ID	City	Degree	Age	Salary	Married ?
1	Lisbon	NaN	25	45,000	0
2	Berlin	Bachelor	25	NaN	1
3	Lisbon	NaN	30	NaN	1
4	Lisbon	Bachelor	30	NaN	1
5	Berlin	Bachelor	18	NaN	0
6	Lisbon	Bachelor	NaN	NaN	0
7	Berlin	Masters	30	NaN	1
8	Berlin	No Degree	NaN	NaN	0
9	Berlin	Masters	25	NaN	1
10	Madrid	Masters	25	NaN	1

¿Filas luego columnas
ó
Columnas luego filas?



Solución 2

En columnas donde el % de NaNs es relativamente bajo, es aceptable reemplazar los NaNs por la media o mediana de la columna.

$$\text{Average_Age} = 26.0$$

ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	NaN	0
7	Berlin	30	1
8	Berlin	NaN	0
9	Berlin	25	1
10	Madrid	25	1



ID	City	Age	Married ?
1	Lisbon	25	0
2	Berlin	25	1
3	Lisbon	30	1
4	Lisbon	30	1
5	Berlin	18	0
6	Lisbon	26	0
7	Berlin	30	1
8	Berlin	26	0
9	Berlin	25	1
10	Madrid	25	1

Solución avanzada

Las técnicas mencionadas producen distorsiones en la distribución conjunta del vector aleatorio. Estas distorsiones pueden ser muy considerables y afectar en gran medida el entrenamiento del modelo. Para reducir este efecto se puede utilizar **MICE (Multivariate Imputation by Chained Equation)**

1. Se trata cada columna con missing values como la variable dependiente de un problema de regresión.
2. Se van haciendo los fits de cada columna de manera secuencial.
3. Se utiliza la regresión para completar los missing values.

One hot encoding

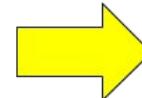
En muchos problemas de Machine Learning, puedo tener como dato de entrada variables categóricas. Por ejemplo, una columna con información sobre el color: {rojo, amarillo, azul}

Para este tipo de información, donde no existe una relación ordinal natural entre las categorías, no sería correcto asignar números a las categorías.

Una forma más expresiva de resolver el problema es utilizar “one hot encoding” y transformar la información en binaria de la siguiente manera.

$$\mathbb{C}_K \quad \text{dim} = K \quad \rightarrow \quad \mathbb{R}^K$$

Color
Red
Red
Yellow
Green
Yellow



	Red	Yellow	Green
Red	1	0	0
Yellow	1	0	0
Green	0	1	0
Yellow	0	0	1

Regresión lineal

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$



Consideramos nuestro conjunto de datos (o observaciones) x_1, \dots, x_n con $x_i \in \mathbb{R}^D$ son las mediciones del sistema en $y \in \mathbb{R}$ es el conjunto de respuestas del sistema. Llamaremos a \bar{x} variables regresoras e y variable de respuesta dependiente.

En general en ML, vamos a buscar encontrar relaciones $y \propto \bar{x}$: $y = f(\bar{x}; \theta) + E$.

En regresión buscamos inferir $\hat{y} = \hat{f}(x)$, la precisión de la estimación de y podemos separarla en dos componentes: **reducible** (que depende de los datos) y **irreducible**.

Como queremos mejorar el error reducible, tenemos que optimizar el mismo, para ello vamos a suponer un \bar{x} fijo y f conocida, con esto calculamos el **Error cuadrático medio**:

$$\begin{aligned} \mathbb{E} (y - \hat{y})^2 &= \mathbb{E} (f(x) + \varepsilon - \hat{f}(x))^2 \\ (*) &= \underbrace{\mathbb{E} (f(x) - \hat{f}(x))^2}_{\text{Error reducible}} + \underbrace{\mathbb{E} (\varepsilon)^2}_{\text{Error irreducible}} \end{aligned}$$

(*) Acá los supuestos intermedios para llegar.

La f más sencilla que podemos pensar es una combinación lineal de los parámetros o regresores
 \rightarrow es simple, es barata, es explicable, \sim precisa.

$$\hat{f}(\bar{x}, \bar{\beta}) = \beta_0 + \sum_{i=1}^D \beta_i x_i$$

Supuestos del modelo lineal:

1. los regresores son independientes $\rightarrow P(x_1, x_2, \dots, x_n) = P(x_1) P(x_2) \dots P(x_n)$
2. Ausencia de colinealidad $\rightarrow \nexists (i, j) / \lambda_i x_i + \lambda_j x_j = x_k$
3. El proceso de generación de datos es homocélico \Rightarrow los ϵ_i iid Y NO dependen de los datos
 $\epsilon_i \sim N(0, \sigma^2) \forall i$

Con estos supuestos limitamos la familia de funciones f que modela mi sistema, en particular tenemos 3 métodos super conocidos:

- ① Mean Square error (MSE). \rightarrow Enfoque empírico
- ② Maximum Likelihood (ML) \rightarrow Enfoque probabilístico
- ③ Maximum a posteriori (MAP) \rightarrow Enfoque bayesiano.

MSE:

partimos de un dataset $\mathcal{D} = \{(x_i, y_i) \mid i \in [1, \dots, K] \text{ } x_i \in \mathbb{R}^{m \times 1}\}$

$$E(\beta) = \sum_{n=1}^K (y_n - \hat{f}(\beta_n))^2 = \sum_{n=1}^K \left(y_n - \beta_0 - \sum_{i=1}^m \beta_i x_i \right)^2 \quad (1)$$

a $x_i = [x_1, \dots, x_m]$ le voy a agregar un 1 al inicio para representar a β_0 .
y $x'_i = [1, x_1, \dots, x_m]$, con esto podemos reescribir (1):

$$\begin{aligned} E(\beta) &= \sum_n \left(y_n - \sum_{i=0}^m \beta_i x_i \right)^2 \\ &= (\bar{y} - \bar{x} \bar{\beta})^t (\bar{y} - \bar{x} \bar{\beta})^2 \quad (2) \end{aligned}$$

Vamos a minimizar (2) $\Rightarrow \partial_{\bar{\beta}} E(\bar{\beta}) = 0$

$$\partial_{\bar{\beta}} E = \partial_{\bar{\beta}} [(\bar{y} - \bar{x} \bar{\beta})^t (\bar{y} - \bar{x} \bar{\beta})] = -2 \bar{x}^t (\bar{y} - \bar{x} \bar{\beta}) = 0$$

$$= \bar{x}^t (\bar{y} - \bar{x} \bar{\beta}) = \bar{x}^t \bar{y} - \bar{x}^t \bar{x} \bar{\beta} \quad \bar{x}^t \bar{x}: \text{matriz de diseño}$$

$$\hat{\beta} = (\bar{x}^t \bar{x})^{-1} \bar{x}^t \bar{y} \quad \rightarrow \quad \hat{y} = H y \quad H = X (X^t X)^{-1} X^t$$

la parte más difícil (y costosa) de este approach es calcular $(X^t X)^{-1}$. Si bretodo si $K \gg m$ (y viceversa) Vamos a tener que no existe la inversa. \Rightarrow para estos casos usamos **pseudoinverso**

(B) Método de máxima verosimilitud (Maximum likelihood)

bajo las condiciones que planteamos sobre la regresión lineal estamos diciendo que existe una distribución de Y condicionada para cada x , $p(Y|X=x, \bar{\beta}, \sigma^2)$. Dado cada par $(x_1, y_1), \dots, (x_n, y_n)$ podemos escribir el siguiente modelo:

$$\prod_{i=1}^n p(y_i|x_i, \bar{\beta}, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \beta_0 - \sum_j \beta_j x_j)^2}{2\sigma^2}} \quad \text{①}$$

$\check{L}(\bar{\beta}, \sigma)$

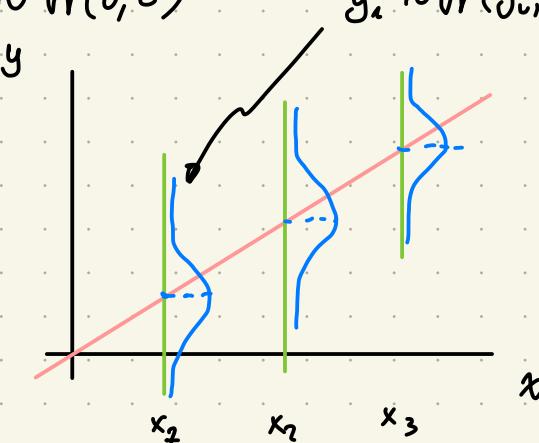
Esta función la conocemos como **función de verosimilitud** de los parámetros y los datos. La forma funcional proviene de propagar la distribución que conocemos ($E_i \sim N(0, \sigma^2)$).

Con esto, lo que vamos a buscar $\max_{\bar{\beta}} \check{L}(\bar{\beta}, \sigma)$:

$\exists \hat{\beta} / \max_{\bar{\beta}} \check{L}(\bar{\beta})$. partimos de $y_i = \hat{f}(x_i, \beta) + \varepsilon$, $\varepsilon \text{ iid } \sim N(0, \sigma^2)$ $\hat{g}_i \sim N(y_i, s)$

$$P(\bar{Y} | \bar{X}, \bar{\beta}) = P(y_1, \dots, y_n | x_1, \dots, x_n, \beta_0, \dots, \beta_m)$$

$$= \prod_{n=1}^{iid} P(y_n | x_n, \bar{\beta}) \sim N(y_n | x_n^T \bar{\beta}, \sigma^2) \quad \text{②}$$



Con esto $\hat{\beta}_{ML} = \arg\max$ (2) :

$$\mathcal{L}(\bar{\beta}) = \arg\max_{\bar{\beta}} \prod_{n=1}^k p(y_n | x_n, \bar{\beta}, \sigma)$$

Si intentamos maximizar esto, vemos rápidamente que puede complicarse rápidamente. Por ello vamos a utilizar el logaritmo de \mathcal{L} para convertir la productoria (\prod) en una sumatoria (\sum):

Log likelihood

$$l(\bar{\beta}) = \log \mathcal{L}(\bar{\beta})$$

$$l(\bar{\beta}) = \mathcal{L}(\bar{\beta}) = \sum_{k=1}^n \frac{1}{2\sigma^2} (y_k - x_k^t \bar{\beta})^2 = \frac{1}{2\sigma^2} (\bar{y} - \bar{x}\bar{\beta})^t (\bar{y} - \bar{x}\bar{\beta})$$

$$l(\bar{\beta}) = \frac{1}{2\sigma^2} \|\bar{y} - \bar{x}\bar{\beta}\|^2 \quad (4)$$

Optimizamos (4) :

$$\partial_{\beta} l(\beta) = 0 \rightarrow \partial_{\beta} \left(\frac{1}{2\sigma^2} (\bar{y} - \bar{x}\bar{\beta})^t (\bar{y} - \bar{x}\bar{\beta}) \right) = 0$$

$$\partial_{\beta} (y^t y - 2y^t x \beta + \beta^t x^t y \beta) = 0$$

$$0 - 2x^t x + 2\beta^t x^t x = 0$$

$$-y^t x + \beta^t x^t x = 0$$

$$\beta^t x^t x = y^t x$$

$$\beta^t = y^t x (x^t x)^{-1}$$

Mejoramos a lo mismo !!

$$\hat{\beta}_{ML} = (x^t x)^{-1} x^t y$$

MAP (Maximum a posteriori) Enfoque Bayesiano

En los métodos que vimos anteriormente no ponemos suposiciones sobre los parámetros θ . El método MAP propone asumir la distribución 'a priori' $p(\theta)$. Esto, restringe los valores que pueden tomar. Vamos a considerar $p(\theta) \sim \mathcal{N}(0, 1)$, esto va a limitar el valor de $\theta \in [-2, 2]$ con alta probabilidad (esto es $\pm 2\sigma_\theta$). Teniendo el dataset (X, Y) , en vez de maximizar la fn. de verosimilitud, vamos a buscar los parámetros θ que maximizan la distribución a posteriori $p(\theta | X, Y)$. Si aplicamos el teorema de Bayes:

Teorema de Bayes:

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

$$P(\theta | X, Y) = \frac{P(Y | X, \theta) P(\theta)}{P(Y | X)}$$

M1

En la ec. M1 vamos a buscar

θ_{MAP} que maximize la distib. a posteriori.

Vamos a utilizar un truco similar al log usado en ML.

$$\log(P(\theta | X, Y)) = \log(P(Y | X, \theta)) + \log(P(\theta)) + \text{cte.} \quad \text{M2}$$

no depende de θ

Para encontrar θ_{MAP} , planteamos:

$$\theta_{MAP} \in \operatorname{argmin} \{-\log P(Y | X, \theta) - \log P(\theta)\}$$

Para esto vamos a considerar:

$$-\partial_{\theta} \log p(\theta | x, y) = -\partial_{\theta} \log p(y | x, \theta) - \partial_{\theta} \log p(\theta)$$

Sabiendo que $p(\theta) \sim \mathcal{N}(\phi, b^2 \mathbb{I})$, $\phi = [0, \dots, 0] \in \mathbb{R}^D$; $b^2 \mathbb{I} = \begin{bmatrix} b & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b \end{bmatrix}$ podemos obtener:

$$-\partial_{\theta} \log p(\theta | x, y) = \partial_{\theta} \left(\frac{1}{2\sigma^2} (y - \Phi \theta)^T (y - \Phi \theta) + \frac{1}{2b^2} \theta^T \theta + \text{cte} \right) \quad (\text{M3})$$

donde Φ es la matriz de features $[\mathbb{1}^T, \bar{x}] = \begin{bmatrix} 1 & x_1 & \dots & x_n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_D & \dots & x_n \end{bmatrix}$

A partir de (M3):

$$-\partial_{\theta} \log P(\theta | x, y) = \frac{1}{\sigma^2} (\theta^T \Phi^T \Phi - y^T \Phi) + \frac{1}{b^2} \theta^T$$

tomando $-\partial_{\theta} \log P(\theta | x, y) = 0$

$$\frac{1}{\sigma^2} (\theta^T \Phi^T \Phi - y^T \Phi) + \frac{1}{b^2} \theta^T = 0 \implies \theta^T \left(\frac{1}{2\sigma^2} \Phi^T \Phi + \frac{1}{b^2} \mathbb{I} \right) - \frac{1}{\sigma^2} y^T \Phi = 0$$

Continuando:

$$\Theta^t \left(\Phi^t \Phi + \frac{\sigma^2}{b^2} \mathbb{I} \right) = y^t \Phi \Rightarrow \Theta^t = y^t \Phi \left(\Phi^t \Phi + \frac{\sigma^2}{b^2} \mathbb{I} \right)^{-1}$$

Con esto obtenemos el estimador MAP

$$\Theta_{MAP} = \left(\Phi^t \Phi + \frac{\sigma^2}{b^2} \mathbb{I} \right)^{-1} \Phi^t y$$

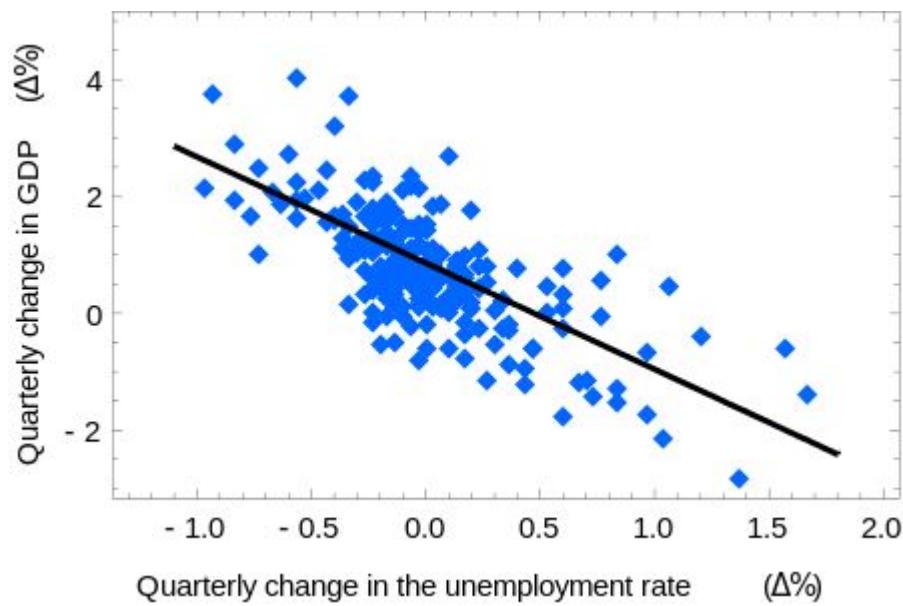
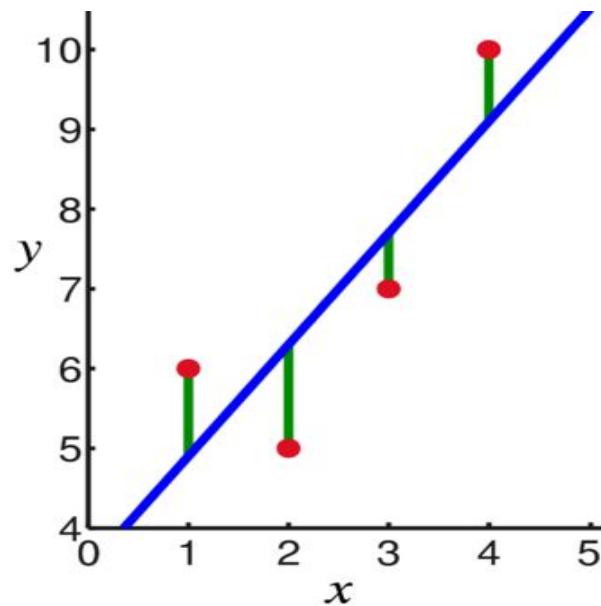
Si vemos el resultado obtenido es muy similar al obtenido previamente salvo por el término $\sigma^2/b^2 \mathbb{I}$. Este término nos asegura que el término a invertir sea simétrico y definido estricto positivo. Esto asegura la existencia de la inversa $\Rightarrow \Theta_{MAP}$ tiene solución única.

Finalmente, Θ_{MAP} tiene un efecto regularizador sobre los parámetros que luego aprovecharemos.

Regresión Lineal

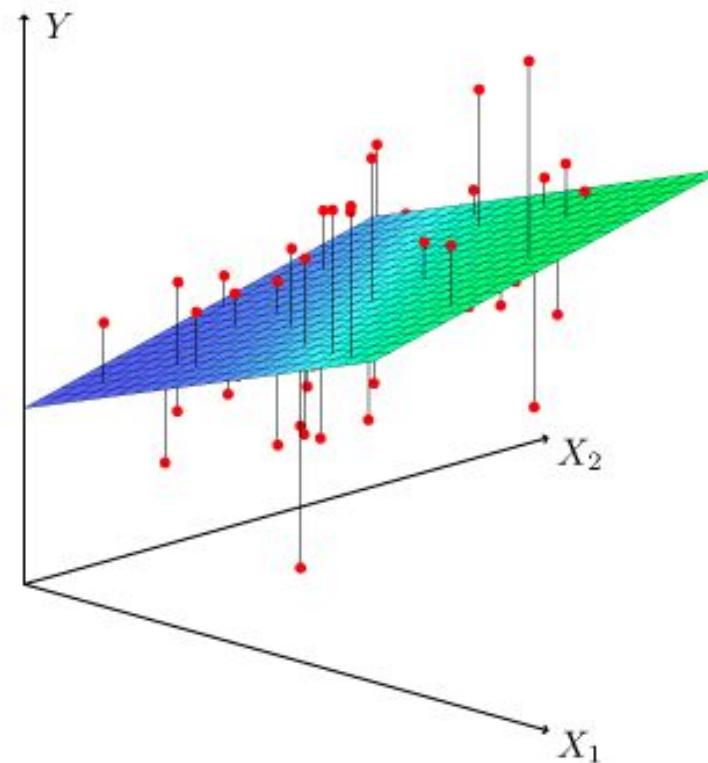
$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

En ésta clase vamos a ver el framework teórico detrás de la gran mayoría de los modelos de Machine Learning: aprendizaje estadístico. Para ello, vamos a utilizar como modelo base la regresión lineal.



Regresión Lineal - Teoría

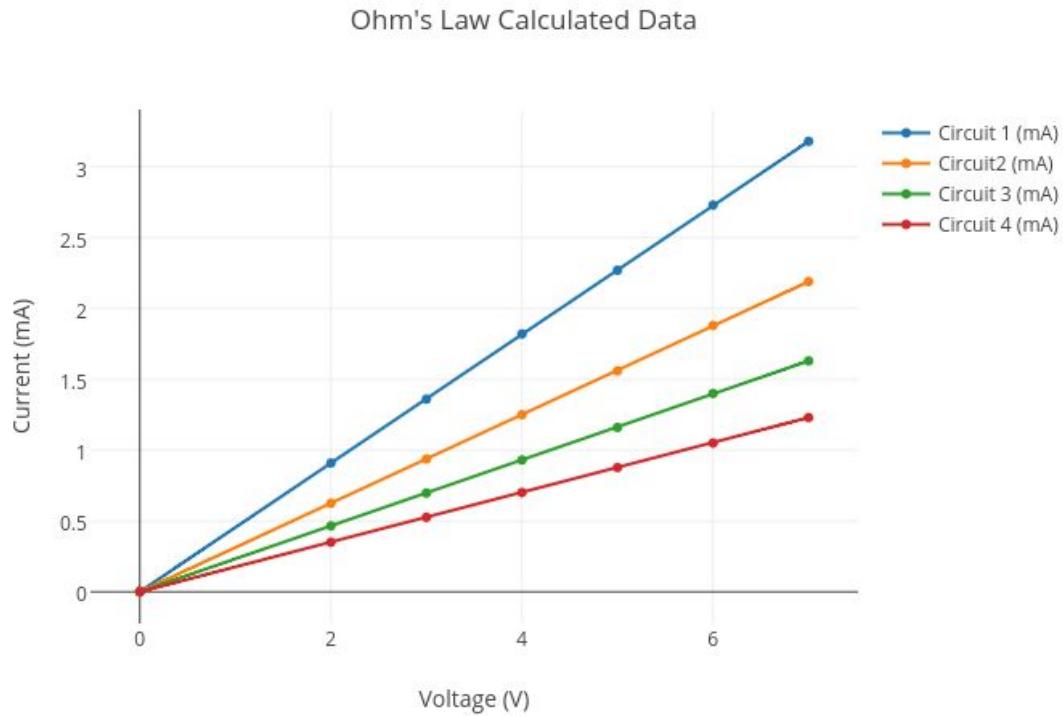
Regresión Lineal $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$



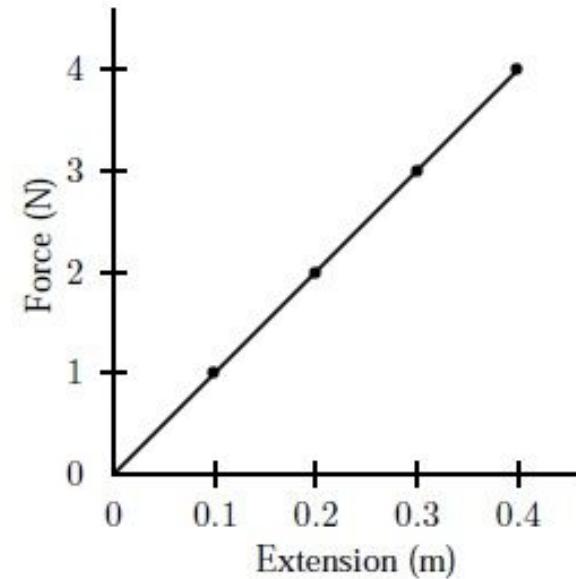
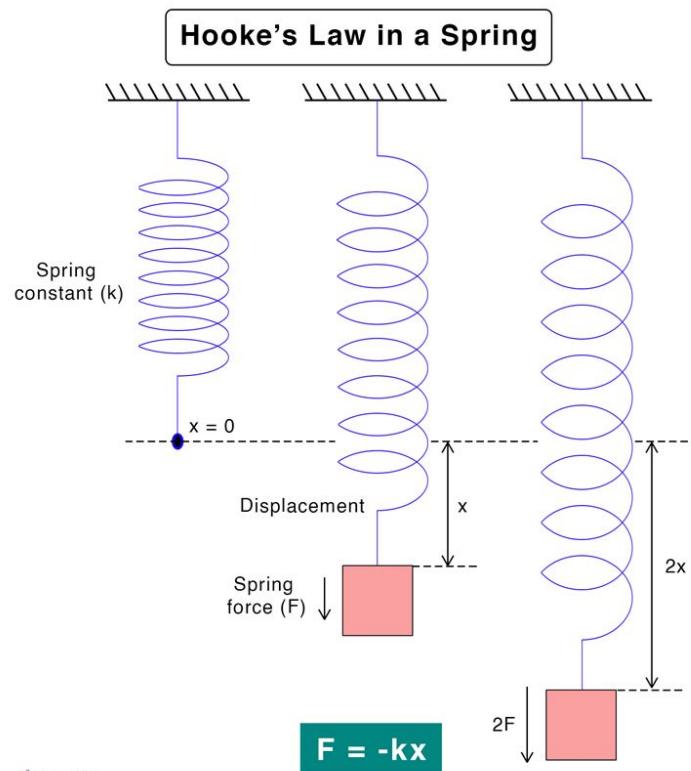
Ley de Ohm

$$I = V/R$$

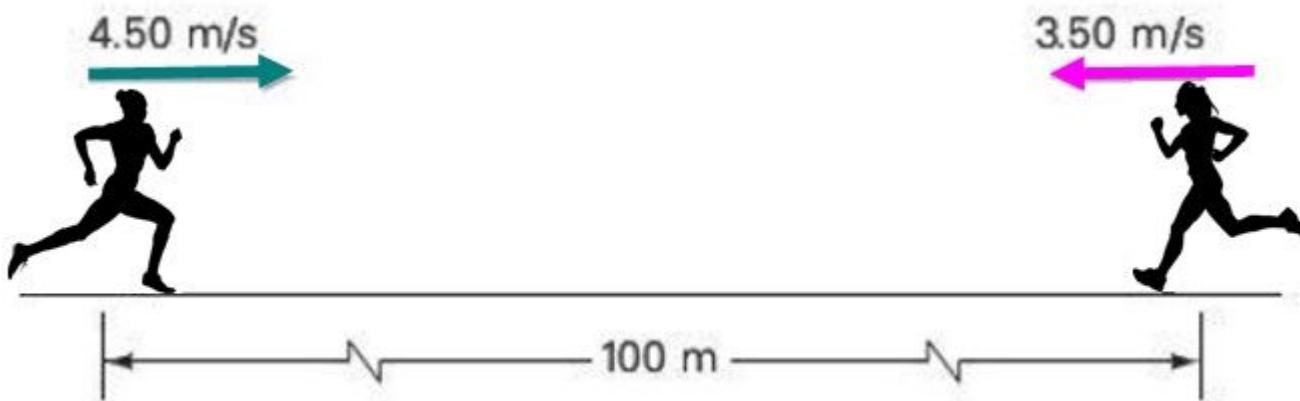
R constante



Ley de Hooke



Movimiento rectilíneo uniforme



$$x(t) = x(t_0) + V * t$$

Población de parásitos



Ejemplo: En un estudio sobre la población de un parásito se hizo un recuento de parásitos en 15 localizaciones con diversas condiciones ambientales.

Los datos obtenidos son los siguientes:

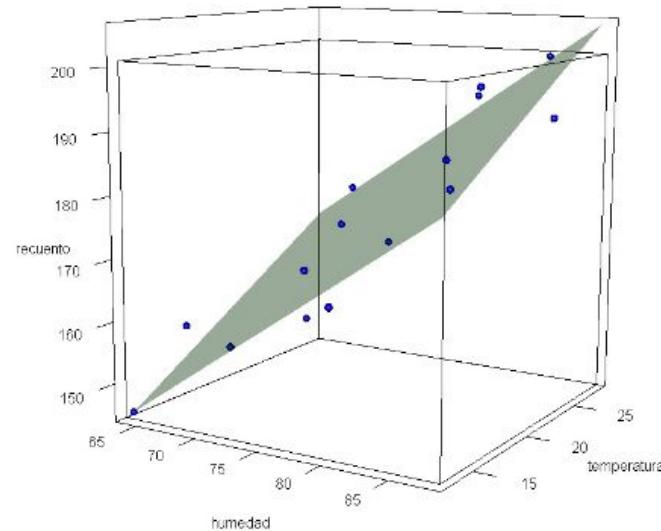
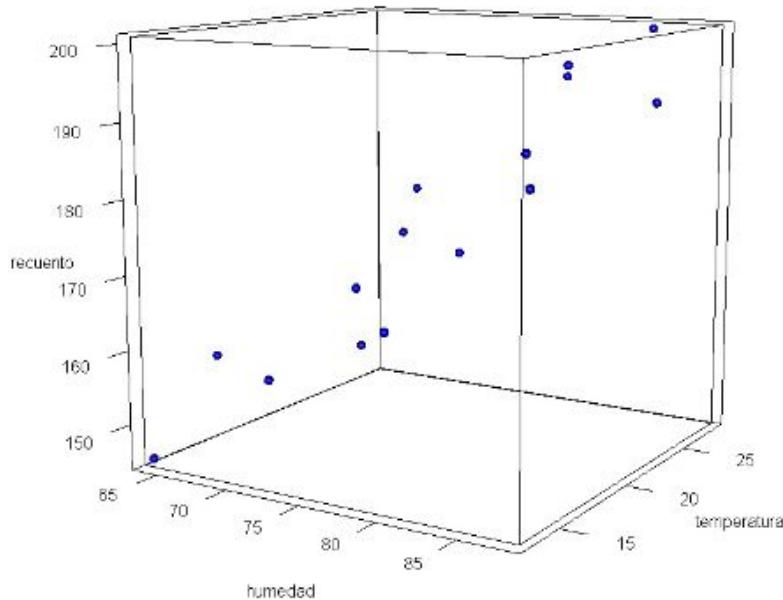
Temperatura	15	16	24	13	21	16	22	18	20	16	28	27	13	22	23
Humedad	70	65	71	64	84	86	72	84	71	75	84	79	80	76	88
Recuento	156	157	177	145	197	184	172	187	157	169	200	193	167	170	192

Fuente:

Población de parásitos

$$\text{Recuento} = \beta_0 + \beta_1 \text{Temperatura} + \beta_2 \text{Humedad} + \epsilon$$

$$\text{Recuento} = 25.7115 + 1.5818 \text{Temperatura} + 1.5424 \text{Humedad}$$



Jamboard



Bibliografía

- The Elements of Statistical Learning | Trevor Hastie | Springer
- An Introduction to Statistical Learning | Gareth James | Springer
- Deep Learning | Ian Goodfellow | <https://www.deeplearningbook.org/>
- Stanford | CS229T/STATS231: Statistical Learning Theory | <http://web.stanford.edu/class/cs229t/>
- Mathematics for Machine Learning | Deisenroth, Faisal, Ong
- Artificial Intelligence, A Modern Approach | Stuart J. Russell, Peter Norvig

