

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



## Machine Learning Terminology

- Raw vs. Tidy Data
  - Training vs. Holdout Sets  $\rightarrow$  holdout (validación, val, Dev)
  - Baseline  $\rightarrow$  baseline  $\Rightarrow$  modelos sencillos
  - Parameters vs. Hyperparameters  $\rightarrow$  hiperparam: la conf. de mi modelo
  - Classification vs. Regression
  - Model-Based vs. Instance-Based Learning
  - Shallow vs. Deep Learning
- lr (learning rate).

clasificación :  $f(\hat{x}, x | y)$  clases con una fn. etiquetadora  $y \in K$

regresión :  $f \propto x, y$  pero la fn  $f(x) \propto y$   $y \in \mathbb{R}$



## Dataset pipeline

Acciones que generalmente se ejecutan sobre los datasets.

Obtención de datos  
o synthetic dataset

Pre-procesamiento  
de Missing Values

Cómputo de media,  
desvío y cuantiles

Estandarización de  
datos (z-score)

Ingeniería de  
Features (PCA)

Data  
augmentation

Split en Train,  
Validation y Test



## Model pipeline

Pasos involucrados al entrenar un modelo de Machine Learning

Obtener el dataset  
para train

Definir métricas de  
evaluación y train

Calcular métricas  
para modelos base

Entrenar el modelo  
con el dataset train

Computar métricas  
con validation

HPs  
optimization

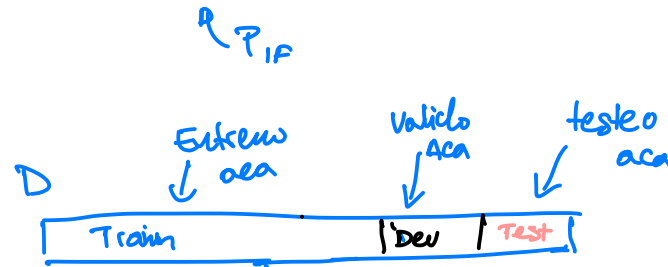
Evaluación sobre  
el dataset test

Handwritten notes:

baseline: RF (random Forest)  
models: ['Knn', 'LDA', 'KSVM']

Handwritten notes:

Esto puede involucrar:  
• Transformaciones  
• Escalados  
• Encoding.



$D_{train}$

$D_{dev}$

$D_{test}$

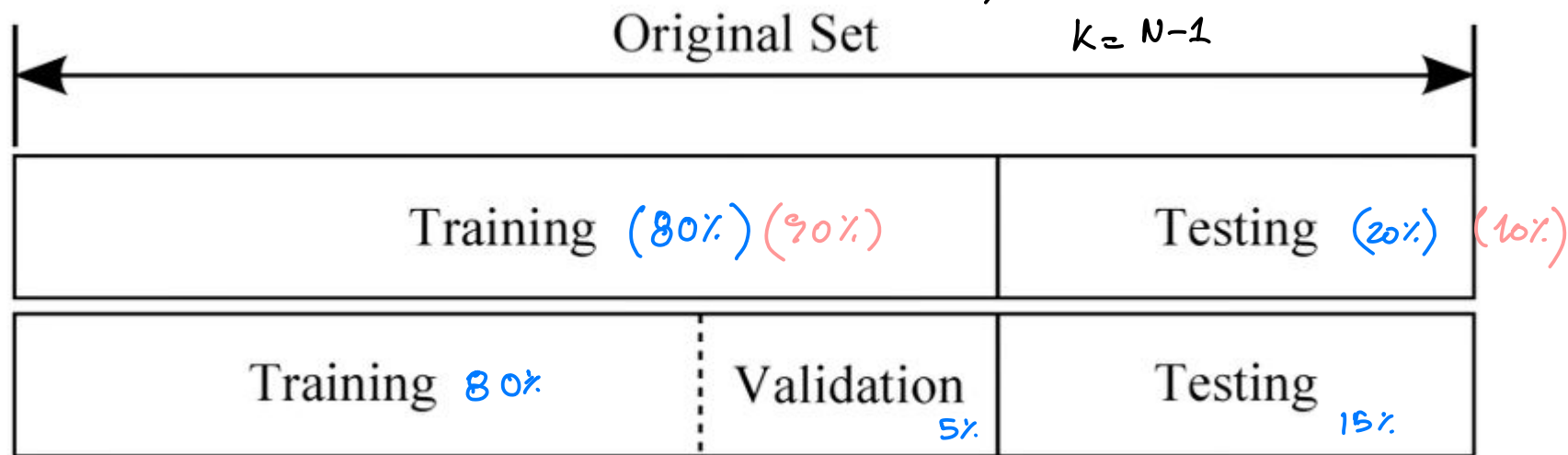




## Ingeniería de Features

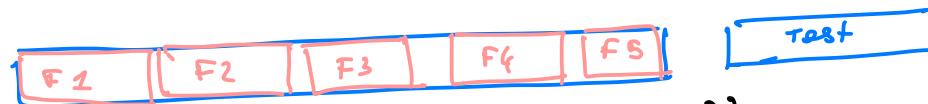
## Train - test - validation

OOT - out of time  
 Leave-one-out  $\rightarrow$  simil (pero para series de tiempo)  
 $\hookrightarrow$  k fold  
 $k = N-1$



training con hold-out

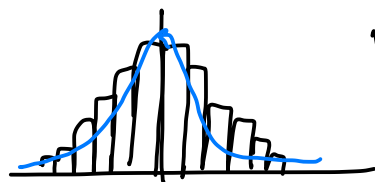
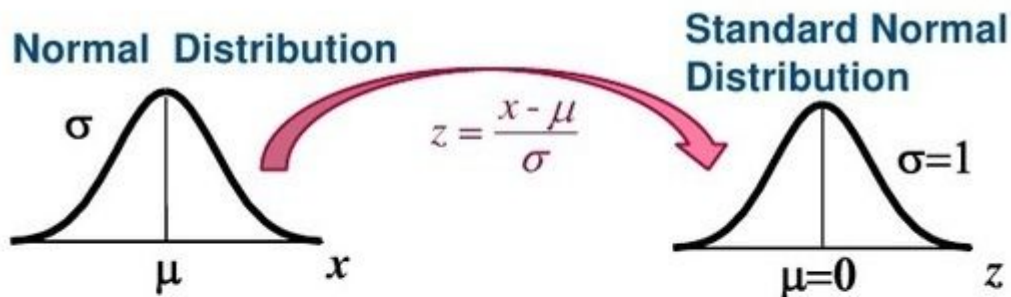
k fold



M1 - Iteración 1 : { Val : F1 , Train : [F2-F5] }

## 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.



no tengo  $\mu, \sigma$   
pero tengo  
 $\bar{x}(\hat{\mu}), s^2(\hat{\sigma}^2)$

$$\rightarrow \hat{z} = \frac{x - \bar{x}}{s}$$



## 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



## 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.

**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.

$$It_0: D_i = D + imp. medias$$

$$It_1: D_1 - D_i = \epsilon_1 \quad \epsilon_n < tol$$

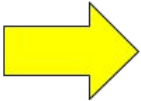
$$It_2: D_2 - D_1 = \epsilon_2$$

## 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.



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

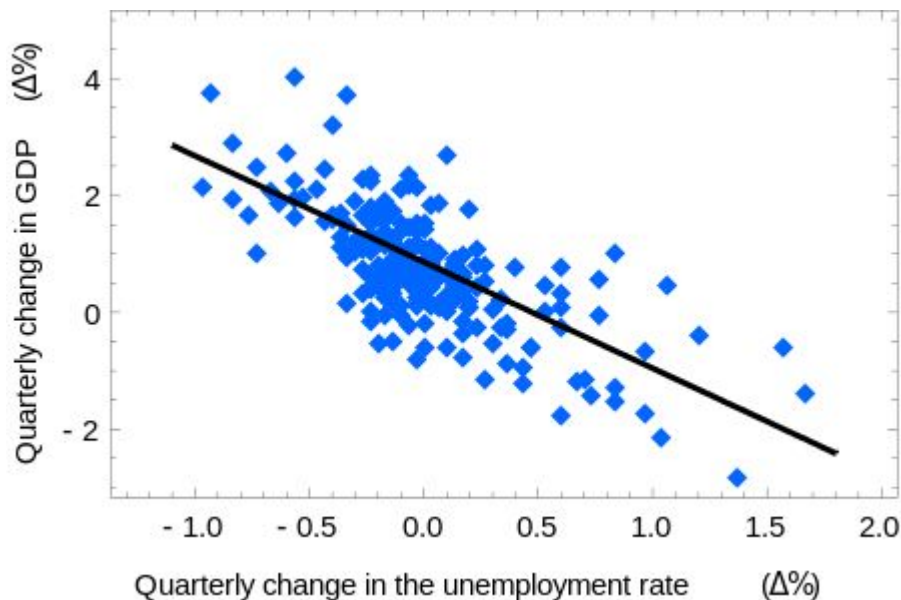
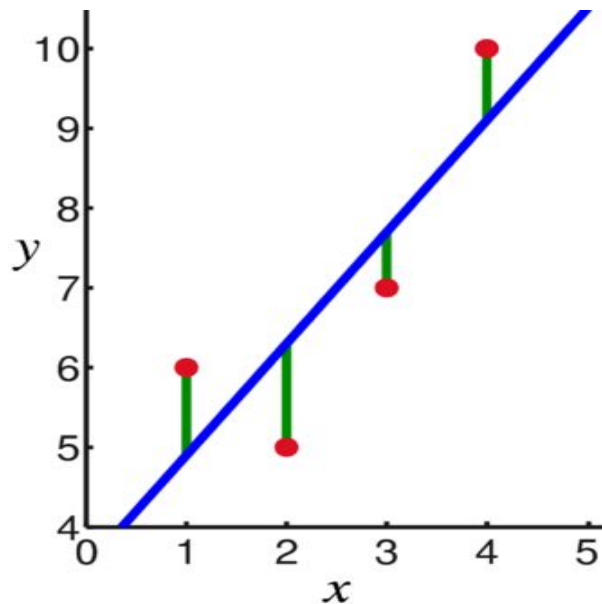
## Regresión lineal

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

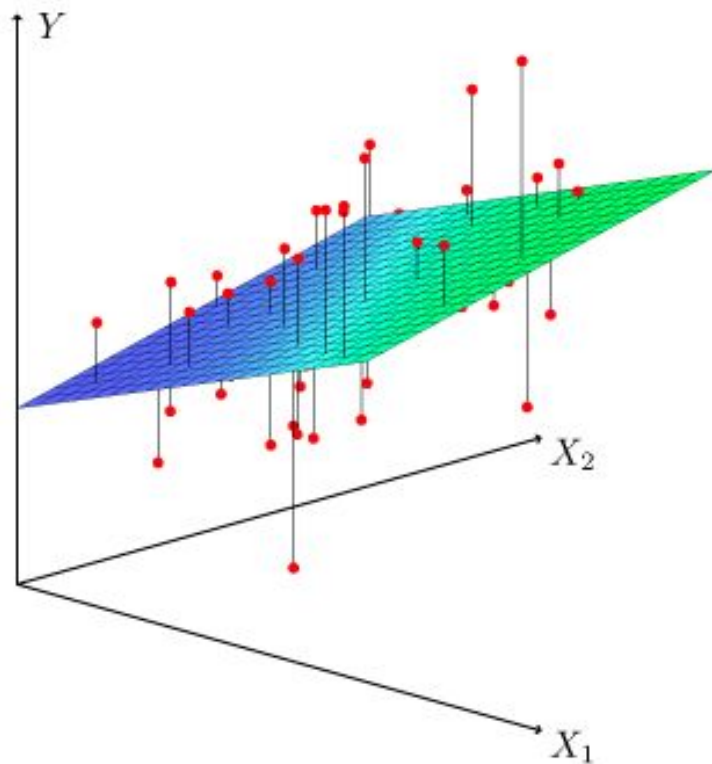


## Regresión Lineal $y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$

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

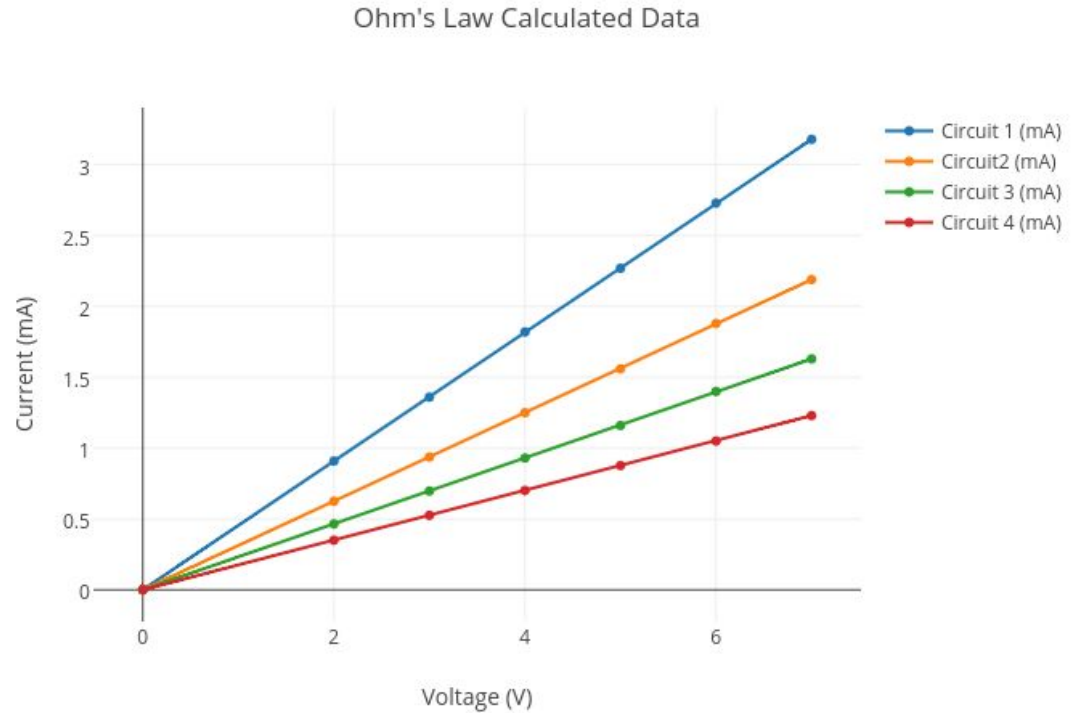




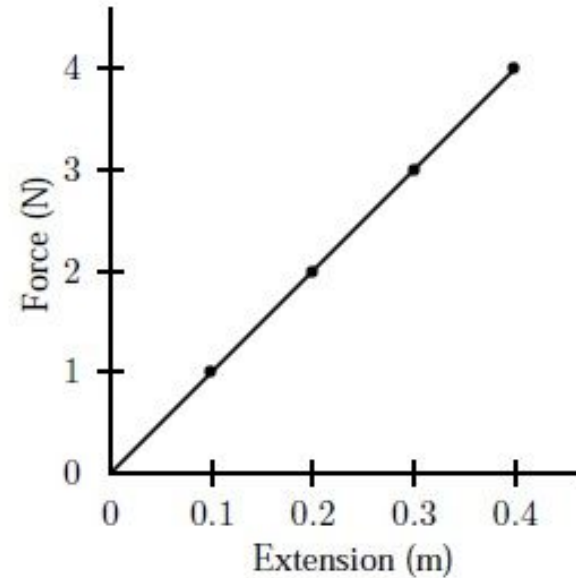
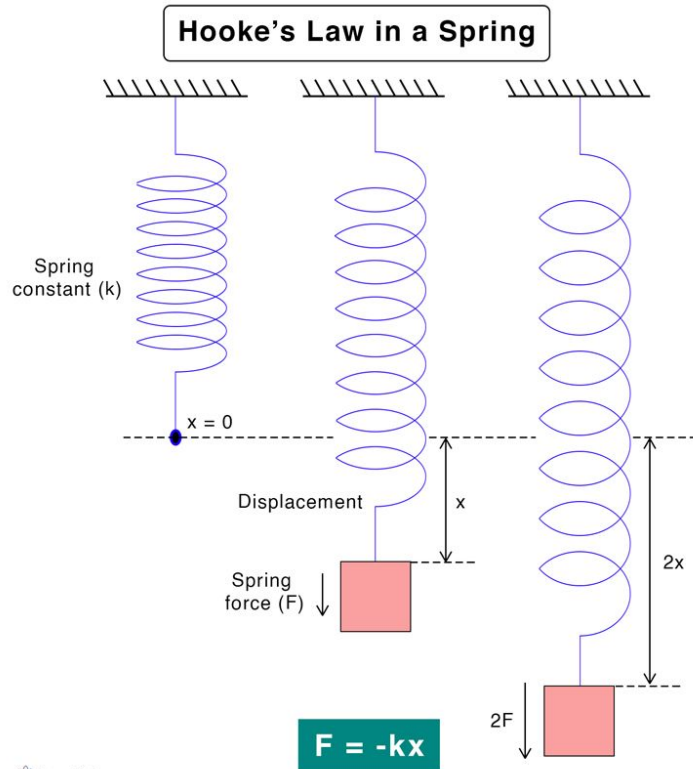
# Ley de Ohm

$$I = V/R$$

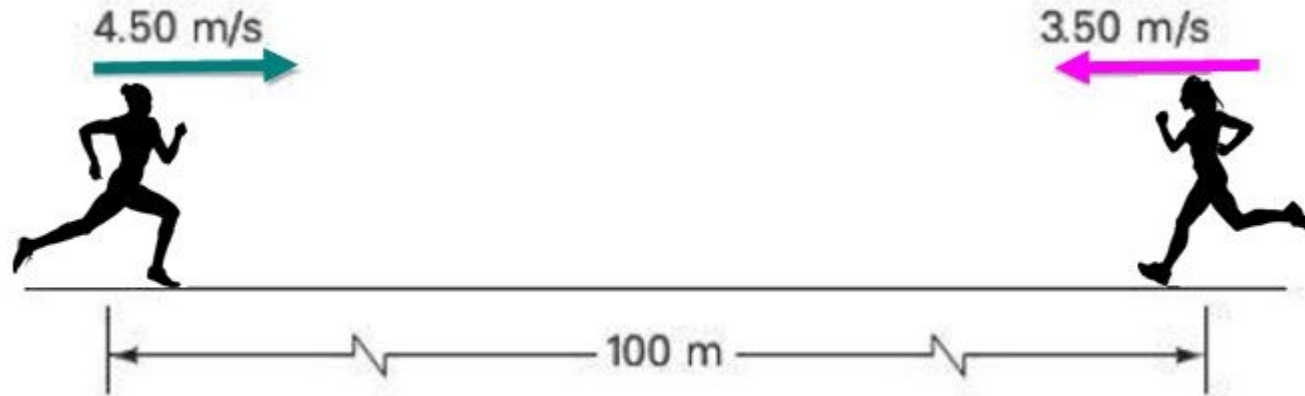
R constante



# 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.

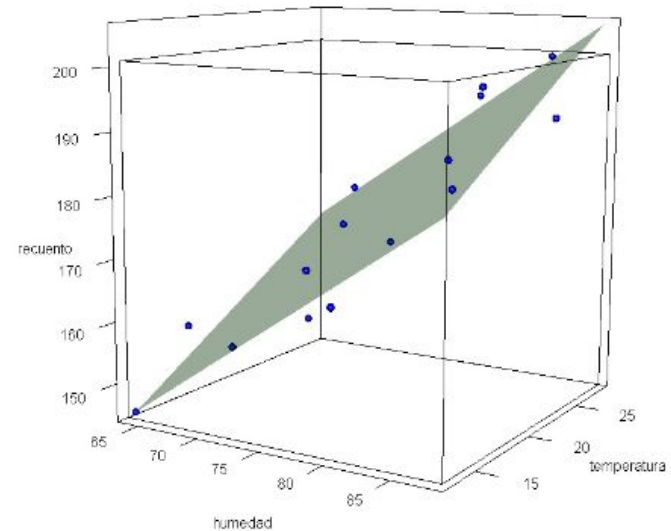
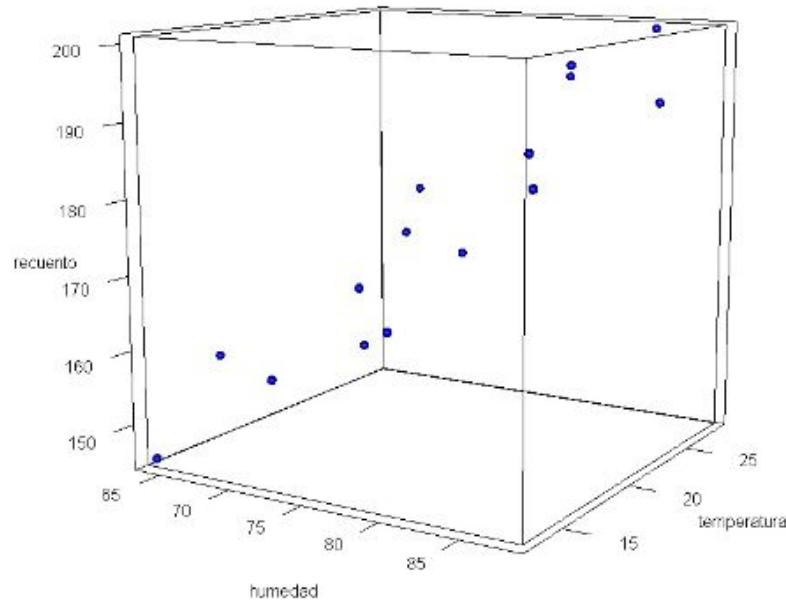
Los datos obtenidos son los siguientes:

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

# 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**



Dados mis datos  $x_1, x_2, \dots, x_n$  con  $x_i \in \mathbb{R}^D$  mediciones de mi sist. e  $y_1, y_2, \dots, y_n$  conjunto de respuestas ( $y_i \in \mathbb{R}$ ). llamamos a  $\bar{X}$  variables regresoras/independientes e  $\bar{Y}$  variable de respuesta dependiente

En yal buscamos encontrar la relación entre  $\bar{X}, \bar{Y}$ :  $y = f(x, \theta) + \varepsilon$

En **regresión** buscamos inferir  $\hat{y} = \hat{f}(x)$ . la precisión de esta medición de  $y$  tiene dos componentes: **reducible** (que depende de los datos) y una comp **irreducible**.

Con esto y asumiendo  $x$  fijo y  $\hat{f}$  conocida, vamos a calcular el **error cuadrático medio** entre  $y$  y  $\hat{y}$ :

$$\begin{aligned} E(y - \hat{y})^2 &= E(f(x) + \varepsilon - \hat{f}(x))^2 \\ &= \underbrace{E(f(x) - \hat{f}(x))^2}_{\substack{\text{comp. reducible} \\ \text{error de estimación}}} + \underbrace{E(\varepsilon)^2}_{\substack{\text{error irreducible} \\ \text{Var}(\varepsilon)}} \end{aligned}$$

la  $f$  más sencilla es una relación lineal  $\rightarrow$  es simple, es barata,  
es explicable,  $\sim$  precisa

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

- Condiciones del modelo lineal  $\rightarrow$  los regresores son independientes
  - $\hookrightarrow$  Ausencia de colinealidad
  - $\hookrightarrow$  los  $\varepsilon_i$  iid  $\varepsilon_i \sim N(0, \sigma)$   
(homocedasticidad)

con estos supuestos límite como encontrar  $\hat{f}$ , vamos a ver  
3 métodos:

- MSE (Mean Square Error)  $\rightarrow$  Enfoque Empírico
- ML (Maximum Likelihood)  $\rightarrow$  Enfoque Probabilístico
- MAP (maximum a priori)  $\rightarrow$  Enfoque Bayesiano



• MSE:

parto de un dataset  $D = \{(\bar{x}_i, y_i) \mid \forall i \in [1, \dots, K] \quad \bar{x}_i \in \mathbb{R}^{m \times 1}\}$

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

al vector  $x_i$  le agrego un 1 para representar a  $\beta_0$ , entonces tenemos  $x_i = [1, x_1, x_2, \dots, x_m]$  y con esto, tomando (1):

$$\begin{aligned} \mathcal{E}(\beta) &= \sum_{n=1}^K \left( y_n - \sum_{i=0}^m \beta_i x_{ni} \right)^2 \\ &= (\bar{y} - \bar{X} \bar{\beta})^t (\bar{y} - \bar{X} \bar{\beta}) \quad (2) \end{aligned}$$

Vamos a tratar de optimizar (2), buscamos  $\partial_{\beta} \mathcal{E} = 0$

$$\begin{aligned} 0 &= \partial_{\beta} \mathcal{E} = \partial_{\beta} [(\bar{y} - \bar{X} \bar{\beta})^t (\bar{y} - \bar{X} \bar{\beta})] = -2 \bar{X}^t (\bar{y} - \bar{X} \bar{\beta}) \\ &= \bar{X}^t (\bar{y} - \bar{X} \bar{\beta}) = \bar{X}^t \bar{y} - \bar{X}^t \bar{X} \bar{\beta} \end{aligned}$$

$X X^t \rightarrow$  matriz de diseño

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



$$\hat{y} = H y$$

$$H = X (X^t X)^{-1} X^t$$

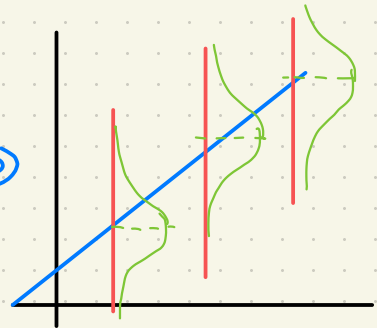
la parte más difícil de esto es obtener  $(X^t X)^{-1}$ . Sobre todo si  $K \gg m$  (y viceversa)  $\leadsto$  para evitar esto se usa *pseudo inversa* (mét. Moore Penrose)

### • Método de Máxima Verosimilitud

$\exists \hat{\beta}$  / maximiza  $\mathcal{L}(\beta)$ . Partimos de  $y_i = \hat{f}(x_i) + \varepsilon$ ,  $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

$$P(y/x|\beta) = P(y_1, \dots, y_n / x_1, \dots, x_n, \beta)$$

$$\stackrel{iid}{=} \prod_{n=1}^K P(y_n / x_n, \beta) \sim \mathcal{N}(y_n / x_n^t \beta, \sigma^2) \quad (3)$$



buscamos  $\hat{\beta}_{ML} = \arg \max (P(\gamma|x, \beta)) :$

$\mathcal{L}(\beta) = \arg \max (P(\gamma|x, \beta))$  si usamos ③  $\mathcal{L}$  es difícil de entender.

si tomamos  $\log \mathcal{L}(\beta) = \ell(\beta)$  convertimos  $\Pi$  en  $\Sigma$

$$\ell(\beta) = - \arg \min_{\beta} (\underbrace{\log (P(\gamma|x, \beta))}_{④})$$

$$④ \log (P(\gamma|x, \beta)) = \frac{1}{2\sigma^2} (y_n - x_n^t \beta)^2 + c$$

$$\begin{aligned} \ell(\beta) &= \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - x_n^t \beta)^2 = \frac{1}{2\sigma^2} \underbrace{(y - X\beta)^t (y - X\beta)}_{\|y - X\beta\|^2} \\ &= \frac{1}{2\sigma^2} \|y - X\beta\|^2 \end{aligned}$$

Varus a optimizer  $\ell$

$$\partial_{\beta} \ell(\beta) = \partial_{\beta} \left( \frac{1}{2\sigma^2} (y - X\beta)^t (y - X\beta) \right) = 0$$

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

$$= 0 - 2y^t X + 2\beta^t X^t X$$

$$= -y^t X + \beta^t X^t X \quad \rightarrow \quad \beta^t X^t X = y^t X$$

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

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

## MAP (Maximum a posteriori) Enfoque Bayesiano

En los métodos que vimos anteriormente no ponemos suposiciones sobre los parámetros  $\theta$ . 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  $\theta$  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  $\theta_{\text{MAP}}$  que maximice la distrib. a posteriori.

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

$$\log(P(\theta/x, y)) = \log(P(y/x, \theta)) + \log(P(\theta)) + \text{cte.}$$

no depende de  $\theta$

(M2)

Para encontrar  $\theta_{\text{MAP}}$ , planteamos:

$$\theta_{\text{MAP}} \in \operatorname{argmin} \{-\log P(y/x, \theta) - \log P(\theta)\}$$

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

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

donde  $\Phi$  es la matriz de features  $[\mathbb{1}^t, \bar{x}] = \begin{bmatrix} 1 & x_{n1} & \dots & x_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{D1} & \dots & x_{nn} \end{bmatrix}$

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

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

Continuando:

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

Con esto obtenemos el estimador MAP

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

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  $\Rightarrow \Theta_{\text{MAP}}$  tiene solución única.

Finalmente,  $\Theta_{\text{MAP}}$  tiene un efecto regularizador sobre los parámetros que luego aprovecharemos.

## 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

