

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 → raw data: dato crudo, en gran data de database
tidy data: data proc. datarehouse sin proc
- Training vs. Holdout Sets → divisiones de los datos para asegurar reproducibilidad
- Baseline → ①
- Parameters vs. Hyperparameters → ②
- Classification vs. Regression → $f_{\gamma/x}(y/x) \vee f_x(x)$
- Model-Based vs. Instance-Based Learning
- Shallow vs. Deep Learning
- LLM → large language model - modelos muy grandes de amplio espectro
- Transformer → Arquitectura de DL que se utiliza como bloques
- Transfer Learning → Proceso de agilizar el entrenamiento mediante fine-tuning específico.
- ① conjunto de modelos super básicos que nos indican un norte.
(Ej modelos de baseline: media, rl, RF, CART, ...)
- ② → shallow Learning → \uparrow Hp ($\sim 20-100$ hp)
- Deep learning → $\uparrow\uparrow\uparrow$ Hp ($\sim 1M$ hp) (~ 1056 hp)



Dataset pipeline

Acciones que generalmente se ejecutan sobre los datasets.

DE → DS (1)

(2)

80
B

20

train

testing

6

80
A

val

5

train

test



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

Feature engineering $\rightarrow f(x; \theta)$
 θ se ajusta en training.
 y se exporta a test, val.

Model pipeline

Pasos involucrados al entrenar un modelo de Machine Learning

(1)

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

(4)

cross validation

1' elección del modelo



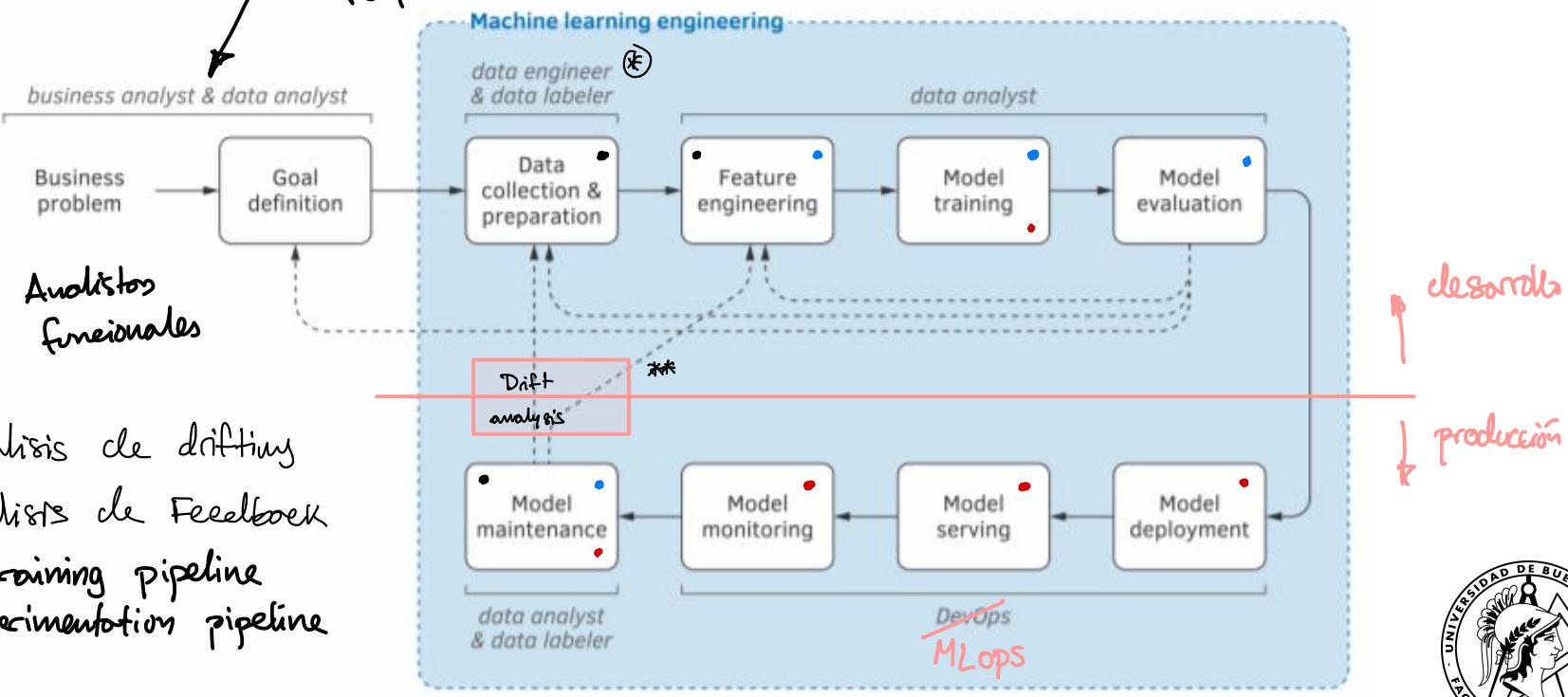
Input Analysis - Machine Learning Pipelines

Machine Learning Pipeline

⊗ DE + convoc. expecto.

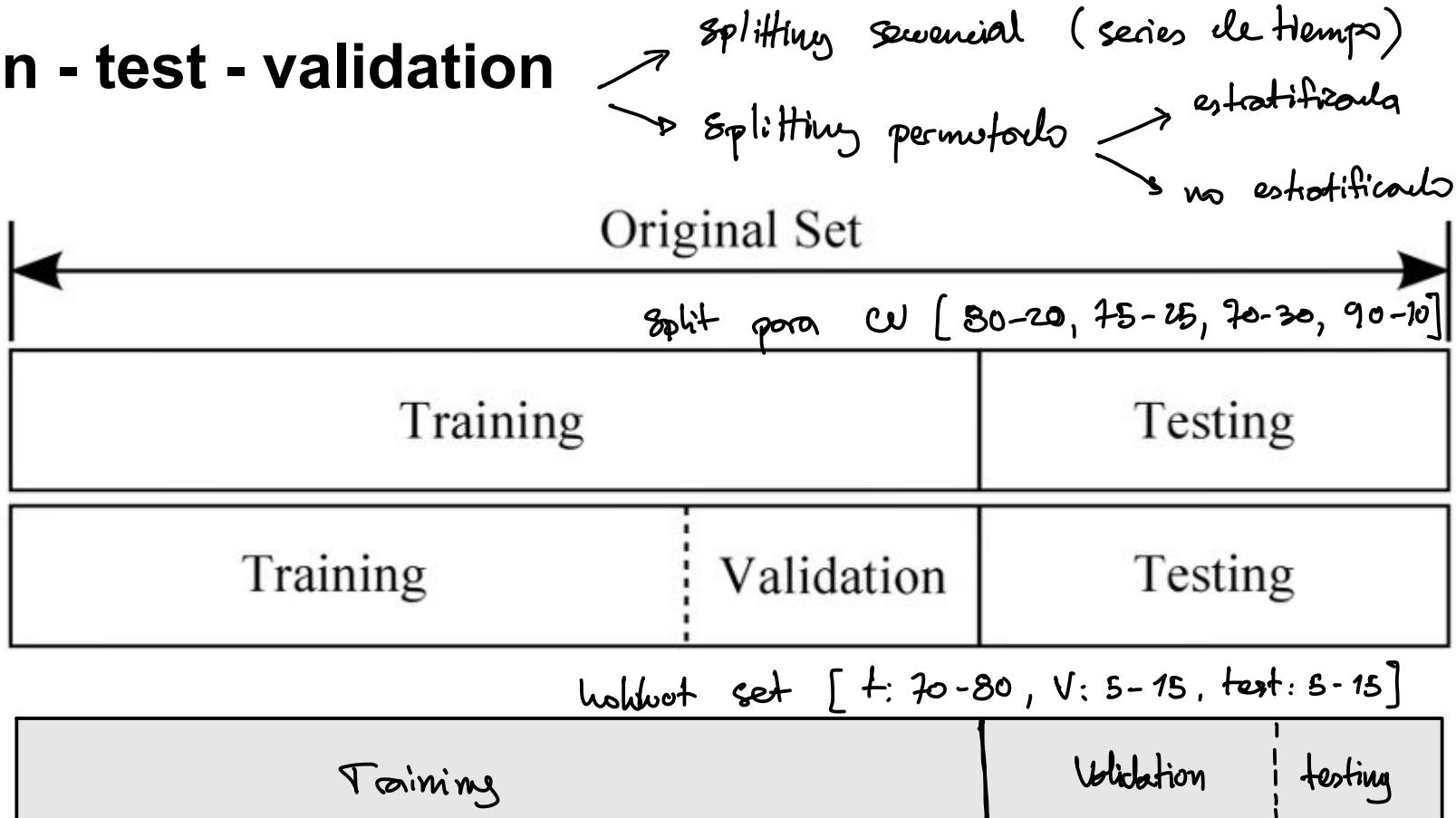
- Data engineer
- Data scientist
- ML engineer (MLOps)

c'Qué queremos
responder?



Ingeniería de Features

Train - test - validation

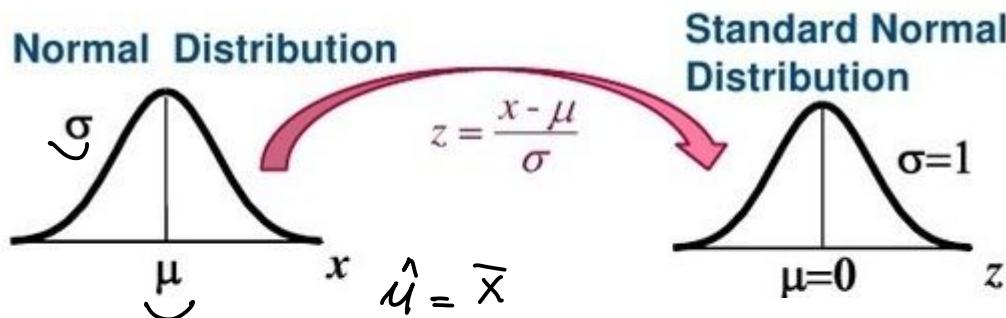


$$f_{train} \approx f_{val} \approx f_{test}$$

80, 13, 5

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.



$$f: \mathbb{R} \mapsto \mathbb{R}$$

$$x \rightarrow z$$

$$\hat{z} = \frac{\bar{x} - \bar{x}}{\hat{s}^2} ; \quad \bar{x} \text{ promedio} \\ \hat{s}^2 \text{ var. muestral}$$

$$N \text{ 95\% } \in [-3\sigma, 3\sigma]$$

$$\phi \text{ 95\% } \in [-3, 3]$$

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

missing values:

- Nan
- inf
- -1
- 999
- -999
- MISSING



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

Complete Case Analysis

¿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.

$\mathcal{C}_k \text{ dim} = k$ $\xrightarrow{\quad} \quad \mathbb{B}^k$

The diagram illustrates the process of one-hot encoding. On the left, a vertical table labeled \mathcal{C}_k shows a categorical column with values: Red, Red, Yellow, Green, Yellow. Above this table is the text $\text{dim} = k$. An arrow points from this table to a large yellow arrow pointing right. To the right of the yellow arrow is a blue matrix labeled \mathbb{B}^k , which represents the binary encoding of the categories. The columns of the matrix are labeled Red, Yellow, Green. The rows correspond to the five entries in the original column. The encoding is as follows: Red is represented by [1, 0, 0], Red is [1, 0, 0], Yellow is [0, 1, 0], Green is [0, 0, 1], and Yellow is [0, 0, 1].

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

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

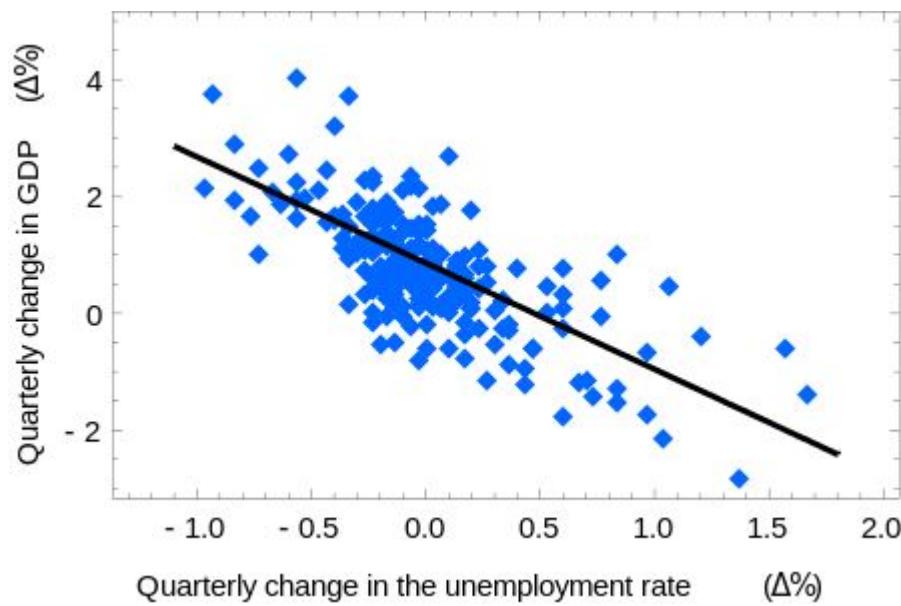
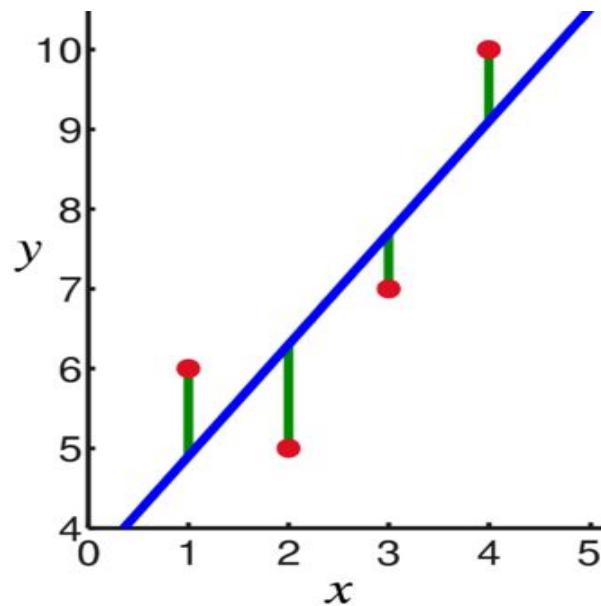
$$\begin{aligned} y &= f(x) + \varepsilon \\ &= \sum_{i=1}^n \omega_i x_i + \varepsilon = x^t \beta + \varepsilon \end{aligned}$$



Regresión Lineal

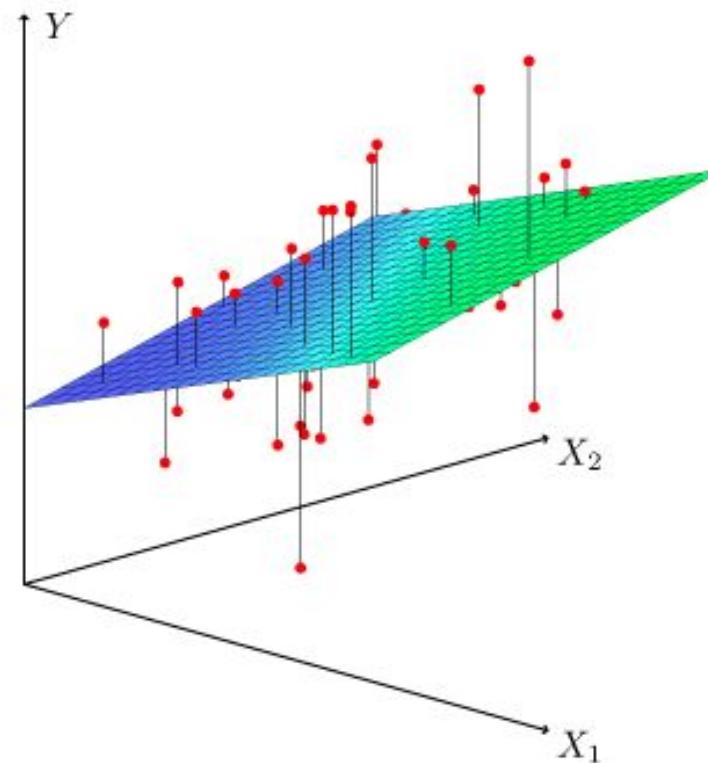
$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$$

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

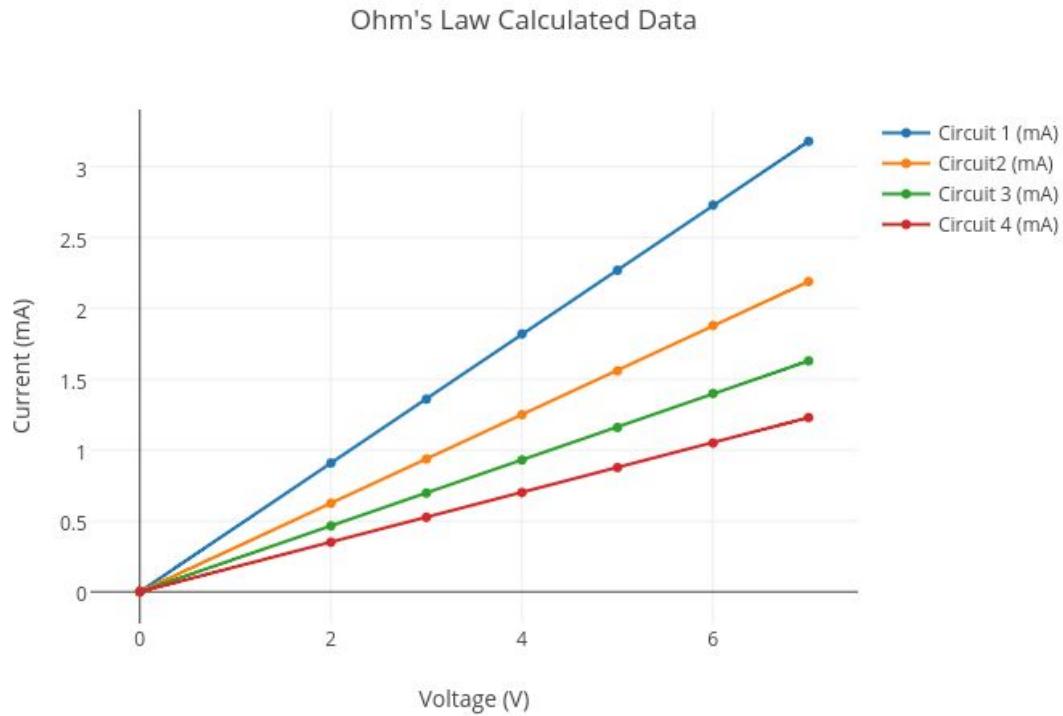
Regresión Lineal $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \dots, n,$



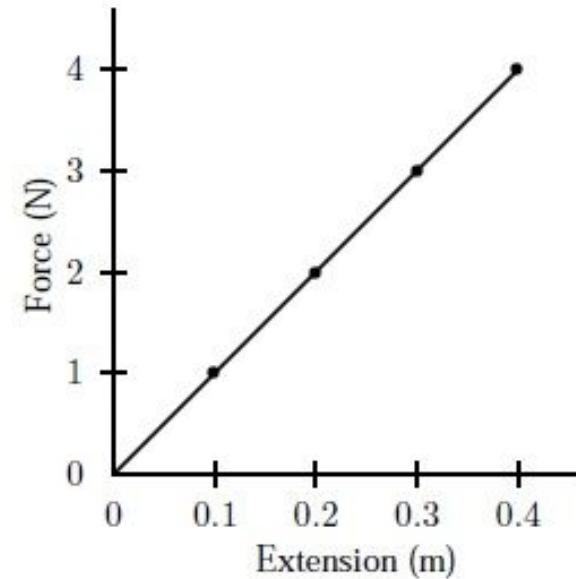
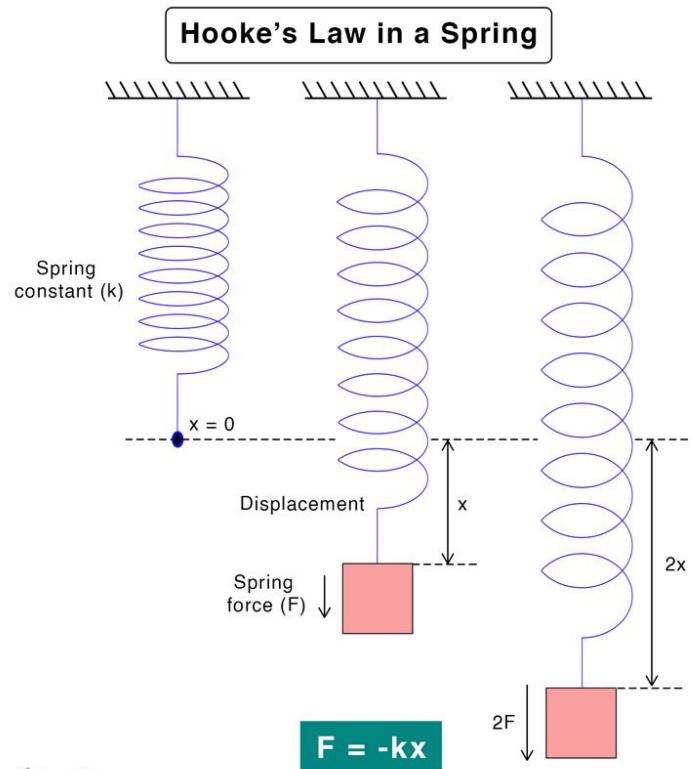
Ley de Ohm

$$I = V/R$$

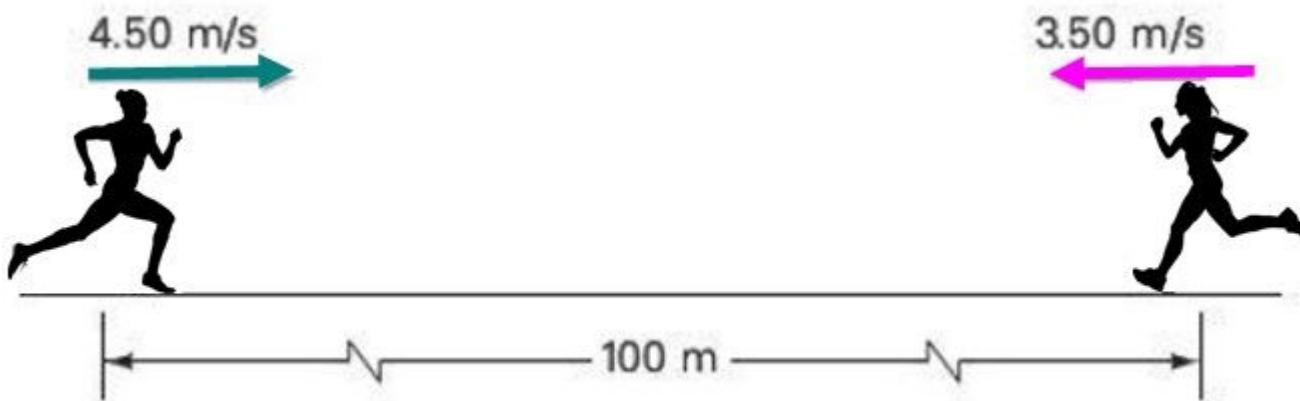
R constante



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

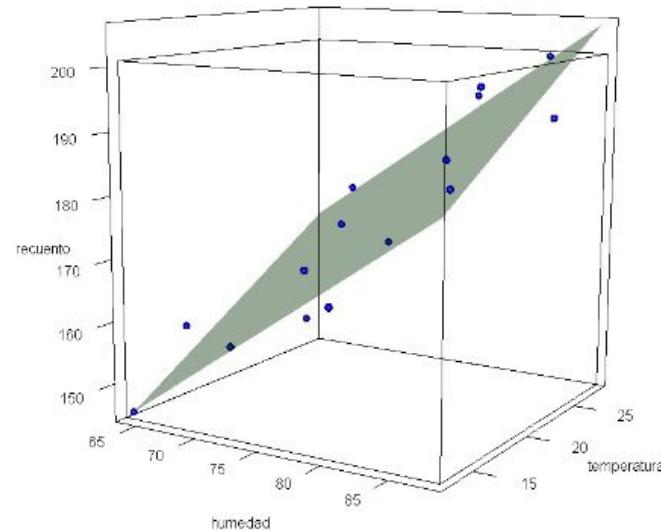
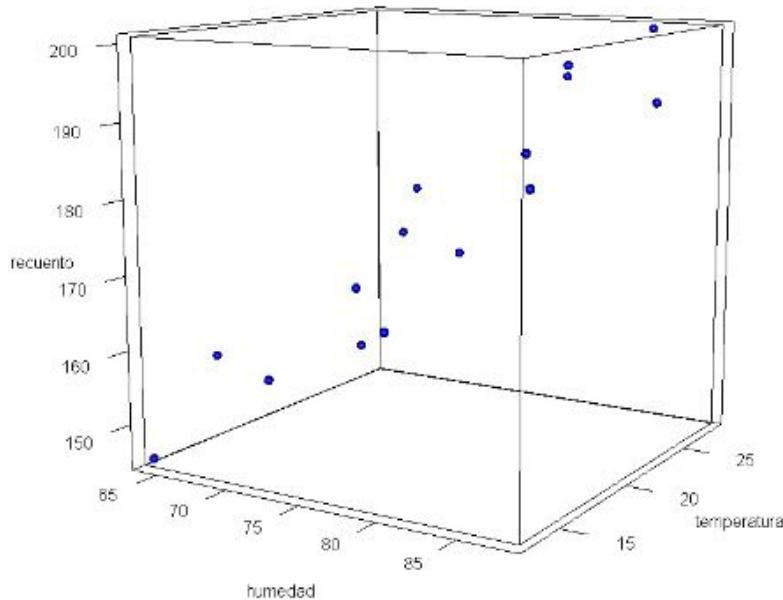
$$\text{Recuento} = \beta_0 + \beta_1 \text{ temp} + \beta_2 \cdot \text{humedad}$$

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



consideremos un conjunto de datos (observaciones) x_1, \dots, x_n con $x_i \in \mathbb{R}^d$ son las mediciones del sistema. Además vamos a considerar $y_i \in \mathbb{R}$ el conj de respuestas del sistema. llamamos a \bar{x} **variables regresoras** e y **variable de resp / dependiente**.

En gral en ml buscamos encontrar una relación entre y & \bar{x} :

$$y = f(\bar{x}; \theta) + \varepsilon \leftarrow \text{comp. aleatorio}$$

en fn de aproximación

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

Como buscamos mejorar el error reducible, tenemos que optimizarlo. Suponemos \bar{x} fijo y f conocida, voy a calcular una forma de error, en particular el **Error cuadrático medio**:

$$\mathbb{E}(\gamma - \hat{\gamma})^2 = \mathbb{E}(f(x) + \varepsilon - \hat{f}(x))^2$$

(*) mediante supuestos.

$$= \underbrace{\mathbb{E}(f(x) - \hat{f}(x))^2}_{\text{error reelucible}} + \underbrace{\mathbb{E}(\varepsilon)^2}_{\text{Error irreelucible}}$$

la f más sencilla que podemos pensar es una comb. lineal de los param. (es simple, es barata, es explicable, \sim precisa)

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

Supuestos del modelo lineal:

0. Existe una relación lineal entre X e Y .
1. Los regresores son independientes $\rightarrow P(x_1, \dots, x_n) = P(x_1) \dots P(x_n)$
2. Ausencia de colinealidad $\rightarrow \beta(i,j), i \neq j / \beta_i x_i + \beta_j x_j = x_k \forall k$
3. El proceso de generación de datos es homocelástico \rightarrow (2)

① \rightarrow los E_i iid γ no dependen de los datos ($E_i \sim N(0, \sigma^2)$)

con estos supuestos limitamos la familia de f 's que modelen el sist. γ con esto podemos tomar 3 métodos:

- ② MSE (Mean Square error) \rightarrow Enfoque Empírico
- ③ ML (Maximum Likelihood) \rightarrow Enf. probabilístico
- ④ MAP (Maximum a posteriori) \rightarrow " bayesiano.

① MSE

partimos de un dataset $D = \{(x_i, y_i) | i \in [1, \dots, K] \quad x_i \in \mathbb{R}^{m \times 1}\}$ y construimos el error:

$$E(\beta) = \sum_{i=1}^K (y_i - \hat{f}(\beta_i))^2 = \sum_{i=1}^K \left(y_i - \beta_0 - \sum_{j=1}^m x_{ij} \cdot \beta_j \right)^2 \quad ①$$

a $x_j = [x_1, \dots, x_m]$ le voy a agregar un 1 como prefijo para representar a β_0 $\Rightarrow x_j^1 = [1, x_1, \dots, x_m]$ con esto:

reescibimos ①:

$$\begin{aligned}\varepsilon(\beta) &= \sum_{i=1}^k \left(y_i - \sum_{j=0}^m x_{ij} \cdot \beta_j \right)^2 \\ &= (\bar{y} - \bar{x}\bar{\beta})^t (\bar{y} - \bar{x}\bar{\beta}) \quad ②\end{aligned}$$

Vamos a minimizar ② $\rightarrow \partial_{\bar{\beta}} \varepsilon(\beta) = 0$

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

$$= \bar{x}^t (\bar{y} - \bar{x}\bar{\beta}) = \bar{x}^t \bar{y} - \underbrace{\bar{x}^t \bar{x}}_{X^t X \text{ matriz de diseño}} \bar{\beta}$$

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

$$\hat{y} = X \hat{\beta} = \underbrace{X}_{H} \underbrace{(X^t X)^{-1}}_{H^{-1}} X^t Y \Rightarrow \hat{y} = H y$$

la parte más difícil (y costosa) de esto es calcular $(X^t X)^{-1}$. Sobre todo si $k \gg m$ (y viceversa) Vemos que no existe la inversa \rightarrow ③

② → para estos casos utilizamos *pseudovolos inversos*

(2) Maximum Likelihood (método de máxima verosimilitud)

bajo las cond. de la reg. lineal estamos diciendo que existe una distrib. de y condicionada para cada x , $P(y/x=x, \beta, \sigma^2)$.

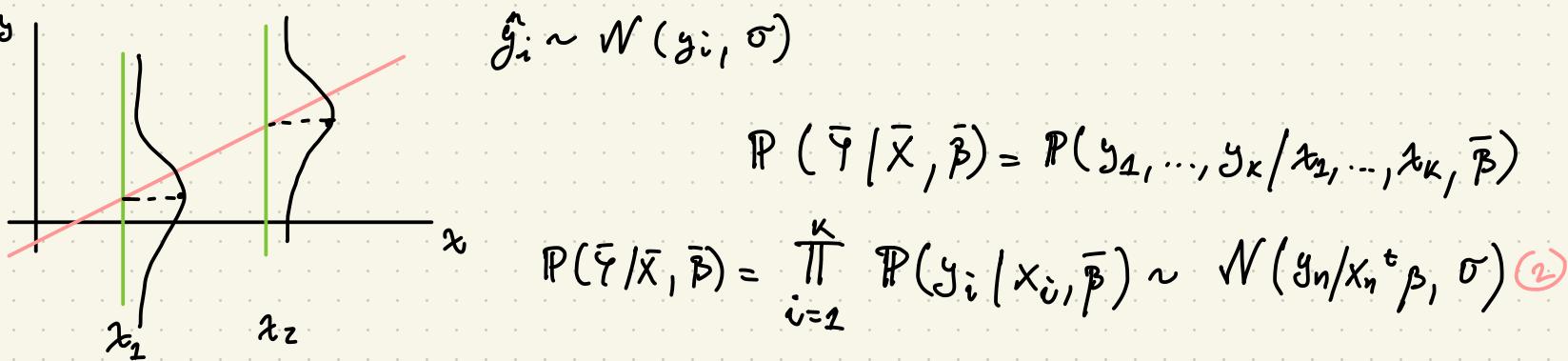
Dado los pares $(x_1, y_1), \dots, (x_K, y_K)$ podemos escribir lo siguiente:

$$\prod_{i=1}^K P(y_i/x_i, \beta, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(y_i - \beta_0 - \sum_j \beta_j x_{ij})^2}{2\sigma^2}} \quad (1)$$

Esta función la conocemos como fn. de verosimilitud $L(\bar{\beta}, \sigma)$ de los parámetros y los datos. La forma funcional proviene de propagar la distrib. que conocemos $\epsilon_i \sim N(0, \sigma^2)$.

Con esto buscamos $\max_{\beta} L$:

$$\exists \hat{\beta} / \max_{\beta} L \rightarrow \text{partimos de } y_i = f(x_i, \beta) + \epsilon$$



$$\hat{y}_i \sim N(y_i, \sigma)$$

$$P(\bar{Y} | \bar{X}, \bar{\beta}) = P(y_1, \dots, y_k | x_1, \dots, x_k, \bar{\beta})$$

$$P(\bar{Y} | \bar{X}, \bar{\beta}) = \prod_{i=1}^k P(y_i | x_i, \bar{\beta}) \sim N(y_i | x_i^T \bar{\beta}, \sigma^2)$$

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

$$\hat{\ell} = \arg \max_{\bar{\beta}} \prod_{i=1}^k P(y_i | x_i, \bar{\beta}, \sigma^2)$$

Si intentamos maximizar $\hat{\ell}$ el problema se torna muy complicado.
Por esto utilizaremos la versión logarítmica (log-likelihood) :

Log likelihood

$$l(\bar{\beta}) = \log \hat{\ell}(\bar{\beta})$$

$$l(\bar{\beta}) = \sum_{i=1}^k \frac{1}{2\sigma^2} (y_i - x_i^T \bar{\beta})^2 = \frac{1}{2\sigma^2} \underbrace{(\bar{y} - \bar{x}\bar{\beta})^T (\bar{y} - \bar{x}\bar{\beta})}_{\|\bar{y} - \bar{x}\bar{\beta}\|^2}$$

$$l(\bar{\beta}) = \frac{1}{2\sigma^2} \|\bar{y} - \bar{x}\bar{\beta}\|^2 \quad \textcircled{3}$$

optimizamos ④:

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

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

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

$$-y^t x + \beta^t x^t x = 0 \rightarrow \hat{\beta}_{ML} = (x^t x)^{-1} x^t y$$

MAP (Maximum a posteriori) Enfoque Bayesiano

En los métodos que vimos anteriormente no ponemos suposiciones sobre los parámetros 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.

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

