



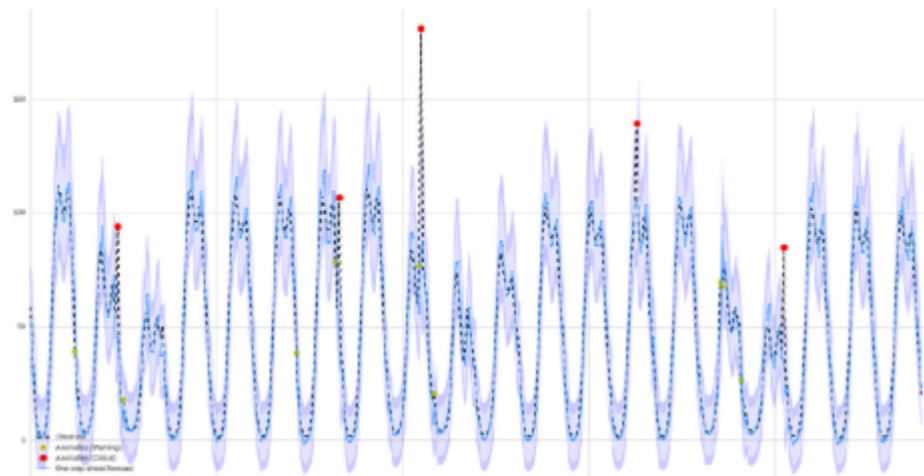
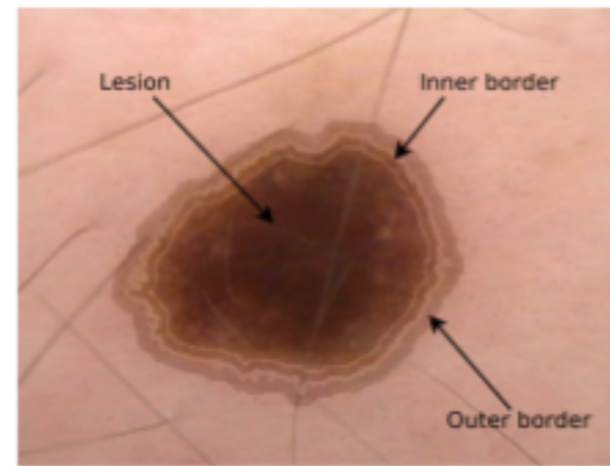
# Clasificación de cargas

# Motivación

- Identificar tipo de electrodoméstico a partir de descriptores derivados de medidas de las señales de voltaje  $V$  y corriente  $I$ .
- Descriptores posibles: potencia activa, reactiva, contenido de armónicos, desfase entre  $V$  e  $I$ , valores de la imagen  $VI$ .
- Realizar una identificación automática:
  - Diseñar un clasificador que aprenda a partir de ejemplos y clasifique electrodomésticos no vistos previamente.
  - Medir el desempeño del sistema.
  - Reportar resultados

# Algunos temas relacionados...

- Reconocimientos de patrones
- Aprendizaje automático
- Inteligencia artificial
- Minería de datos
- Big data analysis
- Clasificación
- Regresión
- Predicción
- Detección de anomalías



- En esta introducción se usan transparencias tomadas del curso:
  - Aprendizaje Profundo para Visión Artificial
- Otros cursos relacionados en el IIE
  - Fundamentos de aprendizaje automático
  - Reconocimiento de Patrones
  - Aprendizaje Profundo por Refuerzo
  - Estimación y Predicción en Series Temporales
  - Tratamiento de imágenes por computadora
  - Procesamiento digital de señales de audio



Electrodoméstico



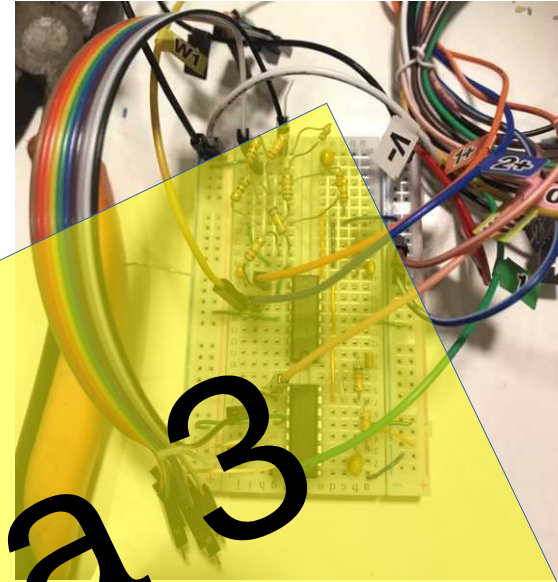
Tipo

- 0: 'Air Conditioner',
- 1: 'Compact Fluorescent Lamp',
- 2: 'Fan',
- 3: 'Fridge',
- 4: 'Hairdryer',
- 5: 'Heater',
- 6: 'Incandescent Light Bulb',
- 7: 'Laptop',
- 8: 'Microwave',
- 9: 'Vacuum',
- 10: 'Washing Machine'

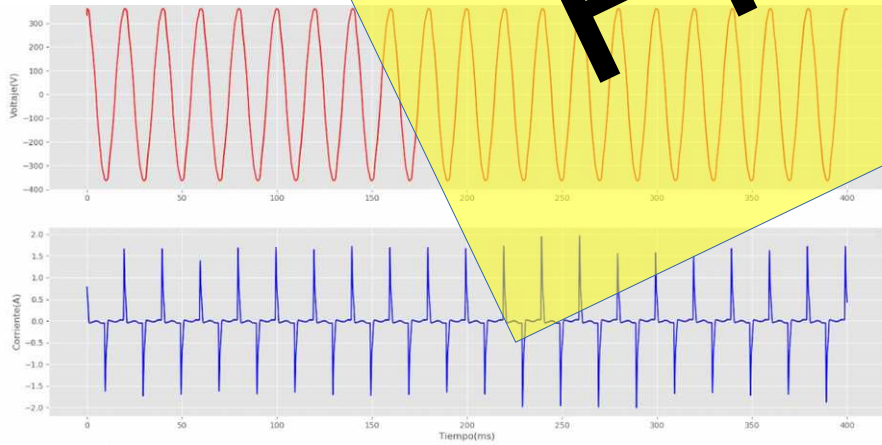


7: 'Laptop'

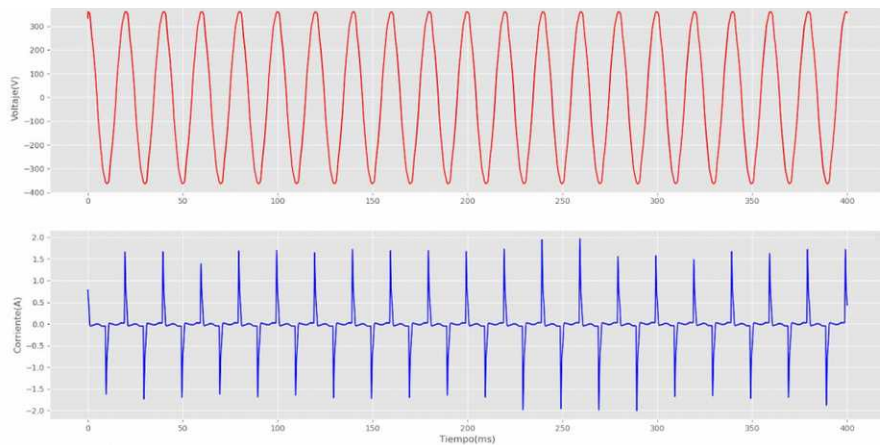
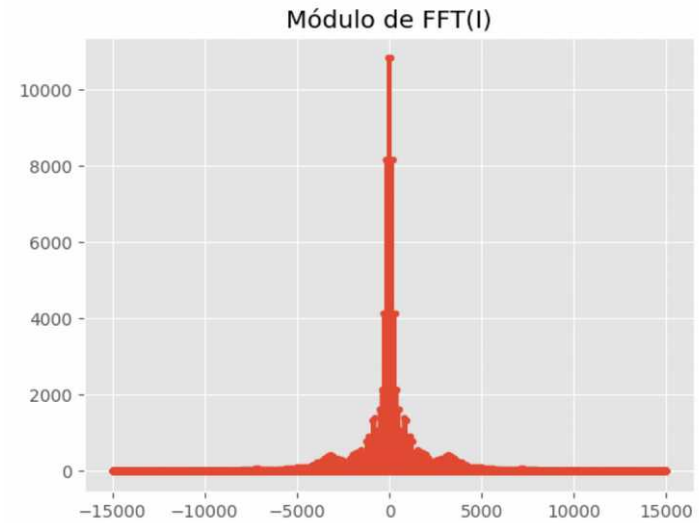
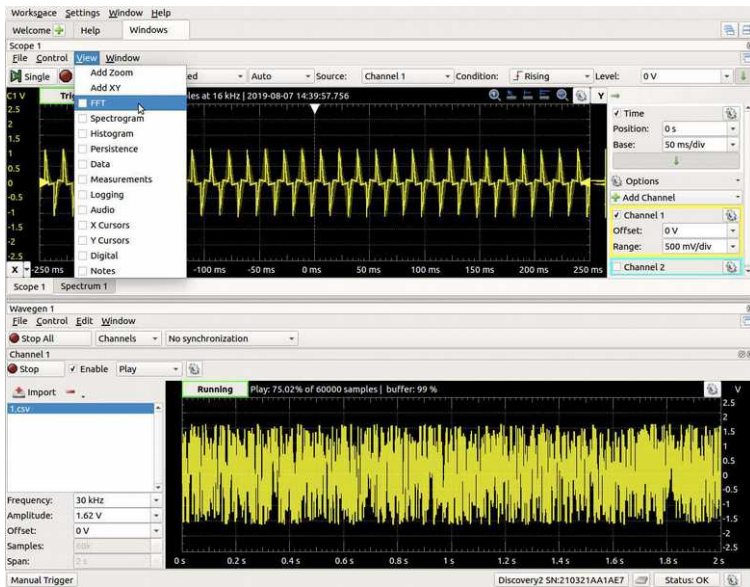
- 0: 'Air Conditioner',
- 1: 'Compact Fluorescent Lamp',
- 2: 'Fan',
- 3: 'Fridge',
- 4: 'Hairdryer',
- 5: 'Heater',
- 6: 'Incandescent Light Bulb',
- 7: 'Laptop',
- 8: 'Microwave',
- 9: 'Vacuum',
- 10: 'Washing Machine'



# Práctica 3



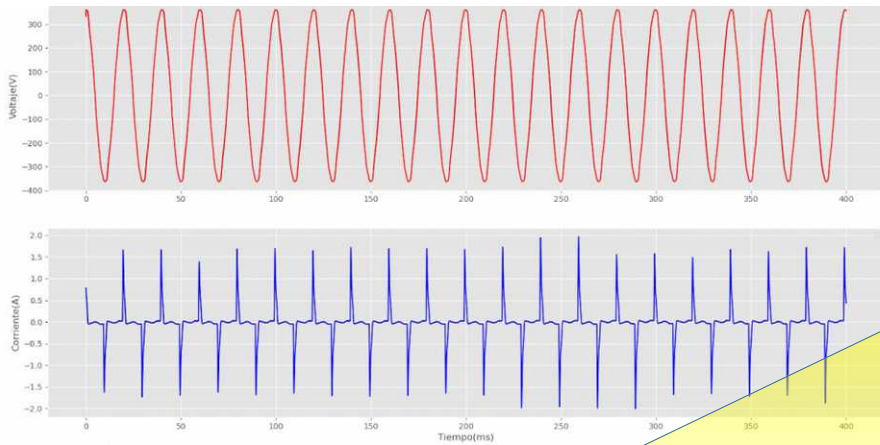




Base PLAID

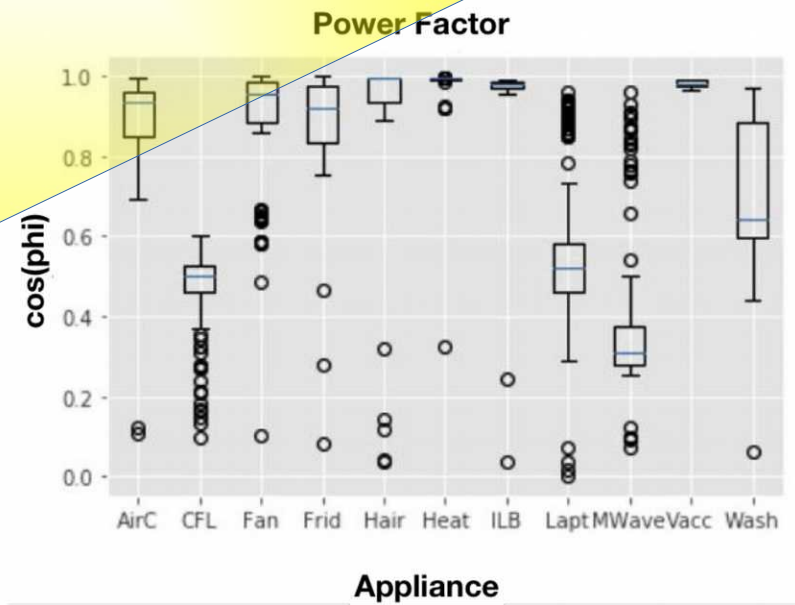
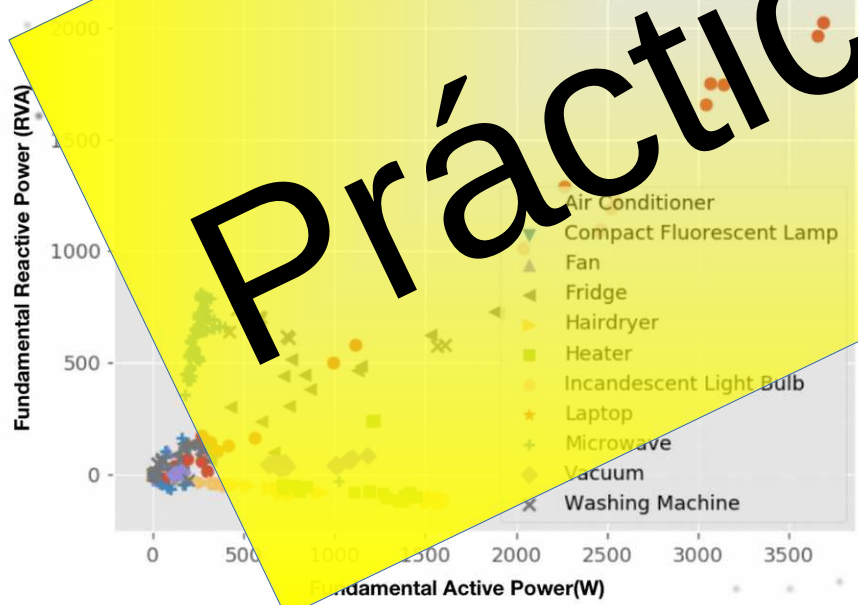
S	Apparent power
S_11	Fundamental apparent power
S_H	Harmonic apparent power
S_N	Non-fundamental apparent power
P	Active power
P_11	Fundamental active power
P_H	Harmonics active power
Q_11	Fundamental reactive power
D_I	Current distortion power
D_V	Voltage distortion power
D_H	Harmonic distortion power
N	Non-active apparent power
THD_V	Total harmonic distortion for voltage
THD_I	Total harmonic distortion for current

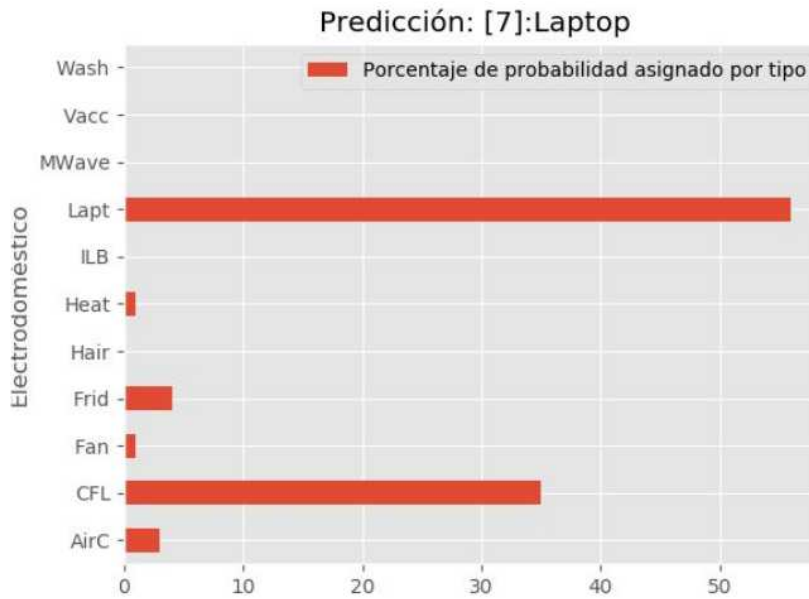
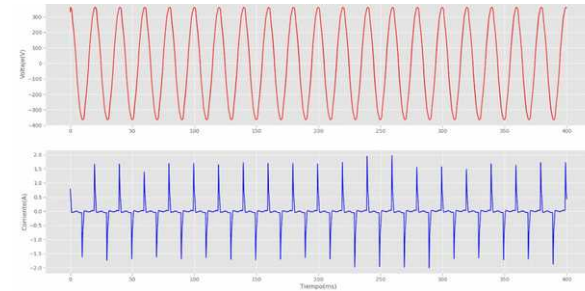
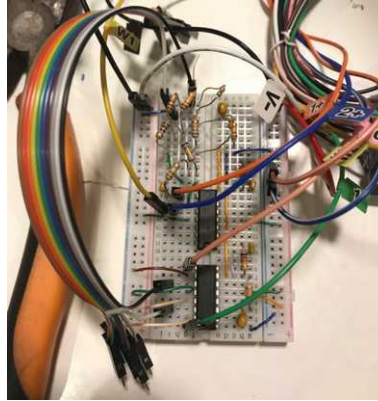




S	Apparent power
S_11	Fundamental apparent power
S_H	Harmonic apparent power
S_N	Non-fundamental apparent power
P	Active power
P_11	Fundamental active power
P_H	Harmonics active power
Q_11	Fundamental reactive power
D_I	Current distortion power
D_V	Voltage distortion power
D_H	Harmonic distortion power
N	Non-active apparent power
THD_V	Total harmonic distortion for voltage
THD_I	Total harmonic distortion for current

# Prácticas 1 y 2

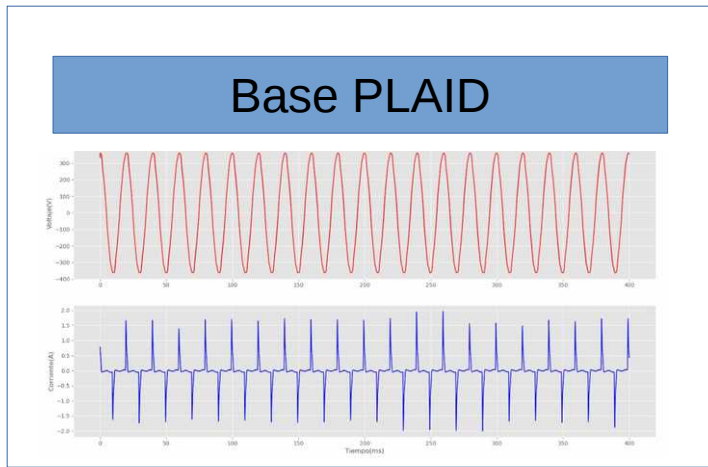




Práctica 4



S	Apparent power
S_11	Fundamental apparent power
S_H	Harmonic apparent power
S_N	Non-fundamental apparent power
P	Active power
P_11	Fundamental active power
P_H	Harmonics active power
Q_11	Fundamental reactive power
D_I	Current distortion power
D_V	Voltage distortion power
D_H	Harmonic distortion power
N	Non-active apparent power
THD_V	Total harmonic distortion for voltage
THD_I	Total harmonic distortion for current



Entrenamiento

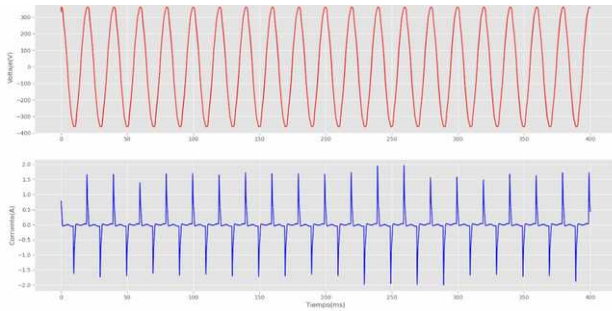
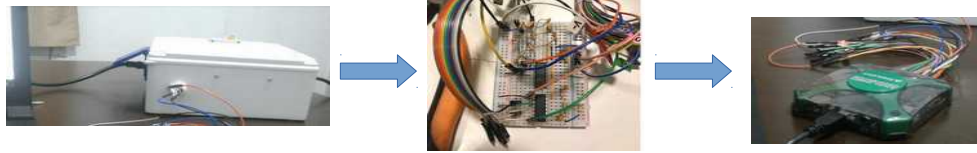
S Apparent power  
 S\_11 Fundamental apparent power  
 S\_H Harmonic apparent power  
 S\_N Non-fundamental apparent power  
 P Active power  
 P\_11 Fundamental active power  
 P\_H Harmonics active power  
 Q\_11 Fundamental reactive power  
 D\_I Current distortion power  
 D\_V Voltage distortion power  
 D\_H Harmonic distortion power  
 N Non-active apparent power  
 THD\_V Total harmonic distortion for voltage  
 THD\_I Total harmonic distortion for current

$\{(x_i, y_i)\}$



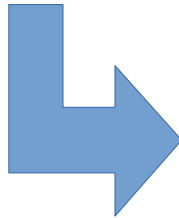
$\{(z_i)\}$   
 z: predicción

x: características  
 y: etiqueta (tipo de edm)

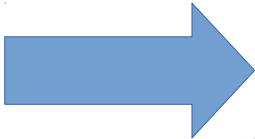


# Funcionamiento en línea

S	Apparent power
S_11	Fundamental apparent power
S_H	Harmonic apparent power
S_N	Non-fundamental apparent power
P	Active power
P_11	Fundamental active power
P_H	Harmonics active power
Q_11	Fundamental reactive power
D_I	Current distortion power
D_V	Voltage distortion power
D_H	Harmonic distortion power
N	Non-active apparent power
THD_V	Total harmonic distortion for voltage
THD_I	Total harmonic distortion for current



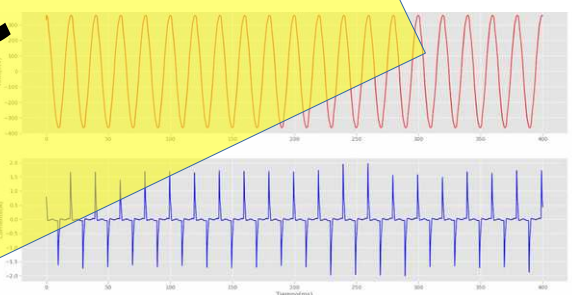
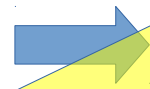
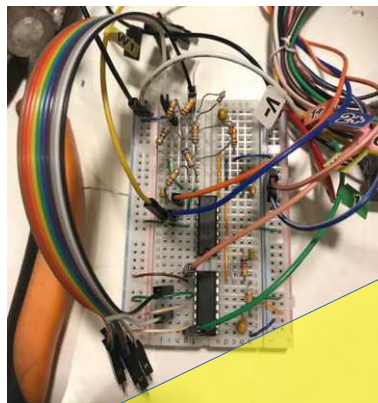
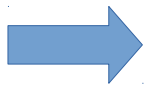
x



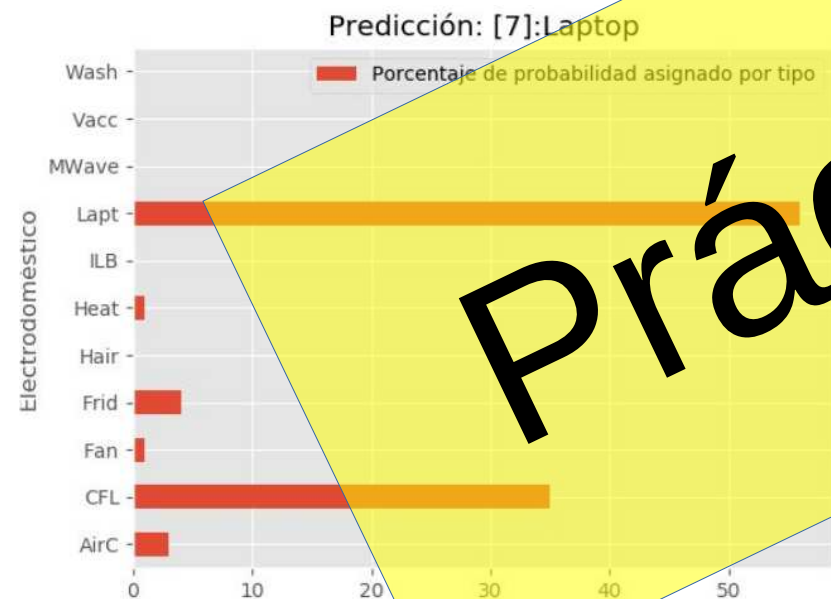
z

z: predicción

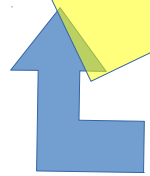
x: características



- S Apparent power
- S\_11 Fundamental apparent power
- S\_H Harmonic apparent power
- S\_N Non-fundamental apparent power
- P Active power
- P\_11 Fundamental active power
- P\_H Harmonics active power
- Q\_11 Fundamental reactive power
- D\_I Current distortion power
- D\_V Voltage distortion power
- D\_H Harmonic distortion power
- N Non-active apparent power
- THD\_V Total harmonic distortion for voltage
- THD\_I Total harmonic distortion for current



# Práctica 5



Clasificador





# Reconocimiento de patrones/Aprendizaje automático





# Reconocimiento de patrones/Aprendizaje automático

El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar objetos, tipos de objetos o relaciones a partir de señales o datos del mundo como usted hace en este momento con estas palabras.



# Reconocimiento de patrones/Aprendizaje automático

“El estudio de cómo las máquinas pueden, observando el ambiente, aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos.”



El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar objetos, tipos de objetos o relaciones a partir de señales o datos del mundo como usted hace en este momento con estas palabras.



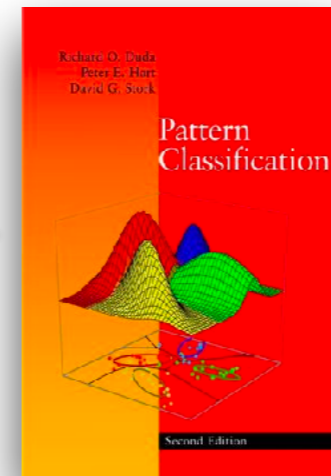
# Reconocimiento de patrones/Aprendizaje automático

“El estudio de cómo las máquinas pueden, observando el ambiente, aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos.”

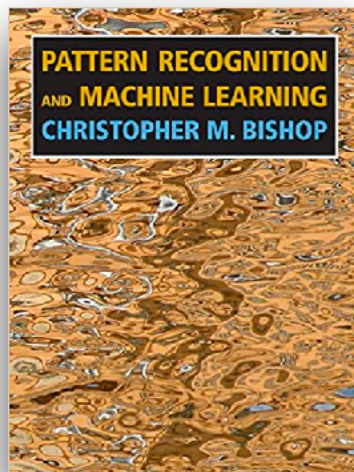


El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar objetos, tipos de objetos o relaciones a partir de señales o datos del mundo como usted hace en este momento con estas palabras.

“El acto de tomar datos crudos y hacer una acción basados en la categoría de los patrones.”



“Descubrir automáticamente regularidades en los datos usando algoritmos en computadora y a partir de esas regularidades tomar acciones.”



# ¿Qué es el aprendizaje automático?

# ¿Qué es el aprendizaje automático?

*“En ciencias de la computación el **aprendizaje automático** es una rama de la inteligencia artificial cuyo objetivo es **desarrollar técnicas** que permitan a las **computadoras aprender**.” (Wikipedia)*

# ¿Qué es el aprendizaje automático?

*“En ciencias de la computación el **aprendizaje automático** es una rama de la inteligencia artificial cuyo objetivo es **desarrollar técnicas** que permitan a las **computadoras aprender**.” (Wikipedia)*

*“A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P** if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.” (Tom Mitchell, Carnegie Mellon University)*

# ¿Qué es el aprendizaje automático?

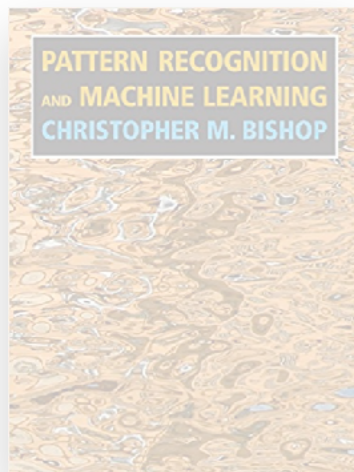
“En ciencias de la computación el **aprendizaje automático** es una rama de la inteligencia artificial cuyo objetivo es **desarrollar técnicas** que permitan a las **computadoras aprender**.” (Wikipedia)

“A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P** if its performance at tasks in **T**, as measured by **P**, improves with experience **E**.” (Tom Mitchell, Carnegie Mellon University)

El **aprendizaje automático** se ocupa de **construir programas** capaces de **mejorar su performance automáticamente** a partir de **experiencia**.

# Reconocimiento de patrones/Aprendizaje automático

“El estudio de cómo las máquinas pueden, observando el ambiente, aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos.”



“Descubrir automáticamente regularidades en los datos usando algoritmos en computadora y a partir de esas regularidades tomar acciones.”

aprendizaje: **mejorar** con la **experiencia** en una **tarea**

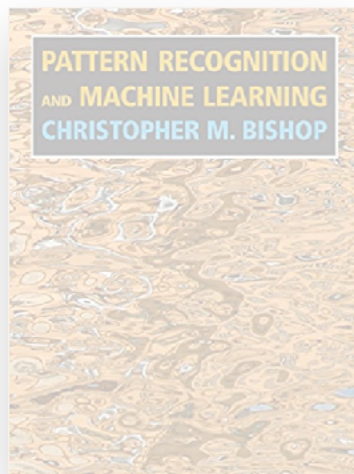
El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar





# Reconocimiento de patrones/Aprendizaje automático

“El estudio de cómo las máquinas pueden, observando el ambiente, aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos.”



“Descubrir automáticamente regularidades en los datos usando algoritmos en computadora y a partir de esas regularidades tomar acciones.”

aprendizaje: **mejorar** con la **experiencia** en una **tarea**

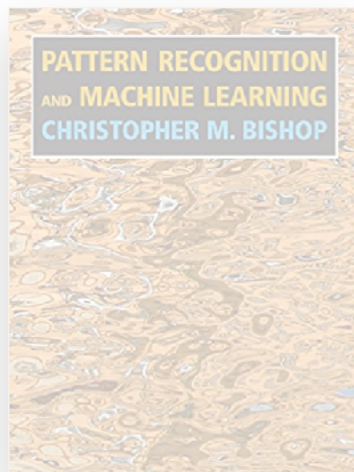
↑  
aplicación

El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar



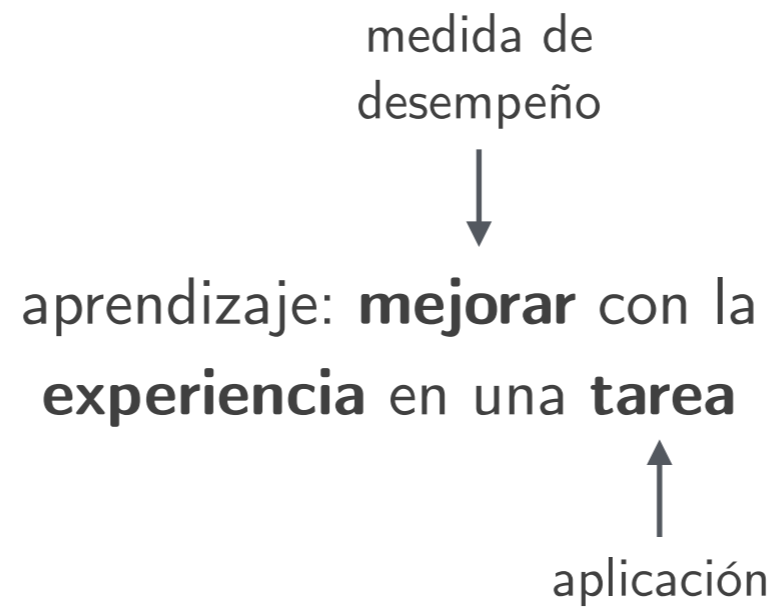
# Reconocimiento de patrones/Aprendizaje automático

“El estudio de cómo las máquinas pueden, observando el ambiente, aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos.”



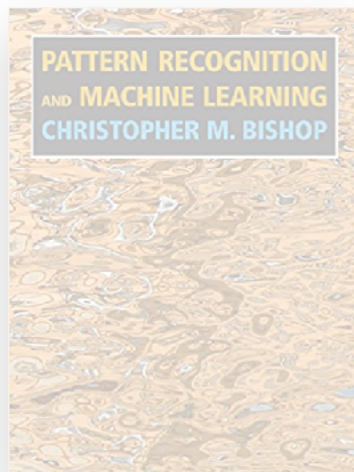
“Descubrir automáticamente regularidades en los datos usando algoritmos en computadora y a partir de esas regularidades tomar acciones.”

El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar



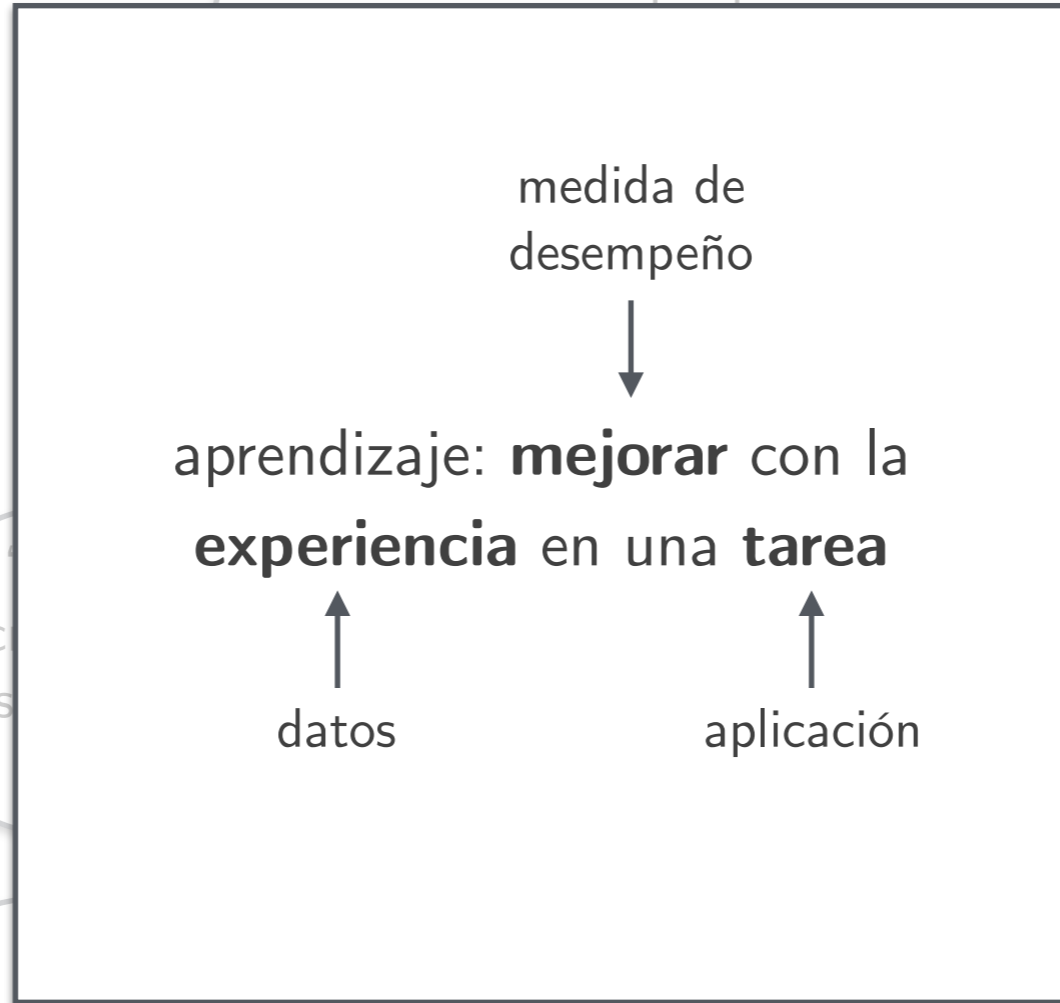
# Reconocimiento de patrones/Aprendizaje automático

“El estudio de cómo las máquinas pueden, observando el ambiente, aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos.”



“Descubrir automáticamente regularidades en los datos usando algoritmos en computadora y a partir de esas regularidades tomar acciones.”

El Reconocimiento de Patrones es una disciplina que estudia métodos que permiten identificar



## 1 Aprendizaje supervisado

- Predecir una salida dada una entrada.
- Regresión (continuo) o Clasificación (discreto)
- Se necesitan datos de entrenamiento con la salida especificada

## 2 Aprendizaje no supervisado

- Encontrar una buena representación de los datos.
- Encontrar estructura, patrones (explorar los datos)

## 3 Aprendizaje por refuerzos

- Aprender a elegir *acciones* con el fin de maximizar alguna noción de recompensa acumulada.
- El programa interactúa con un entorno dinámico

# Formas de aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

# Formas de aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

**Regresión:** La salida deseada es un número real  $y \in \mathbb{R}$  o un vector de números reales  $\mathbf{y} \in \mathbb{R}^m$ .

# Formas de aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

**Regresión:** La salida deseada es un número real  $y \in \mathbb{R}$  o un vector de números reales  $\mathbf{y} \in \mathbb{R}^m$ .

- Predecir la temperatura al mediodía del jueves
- Predecir el valor de la nafta en 6 meses

# Formas de aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

**Regresión:** La salida deseada es un número real  $y \in \mathbb{R}$  o un vector de números reales  $\mathbf{y} \in \mathbb{R}^m$ .

- Predecir la temperatura al mediodía del jueves
- Predecir el valor de la nafta en 6 meses

**Clasificación:** La salida deseada es una variable categorica,  $y \in \{1, \dots, k\}$ .



# Formas de aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

**Regresión:** La salida deseada es un número real  $y \in \mathbb{R}$  o un vector de números reales  $\mathbf{y} \in \mathbb{R}^m$ .

- Predecir la temperatura al mediodía del jueves
- Predecir el valor de la nafta en 6 meses

**Clasificación:** La salida deseada es una variable categorica,  $y \in \{1, \dots, k\}$ .

- El caso más sencillo es clasificación binaria,  $k = 2$ .
- Ejemplos: clasificación de imágenes, reconocimiento de voz.

# Formas de aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

**Regresión:** La salida deseada es un número real  $y \in \mathbb{R}$  o un vector de números reales  $\mathbf{y} \in \mathbb{R}^m$ .

- Predecir la temperatura al mediodía del jueves
- Predecir el valor de la nafta en 6 meses

**Clasificación:** La salida deseada es una variable categorica,  $y \in \{1, \dots, k\}$ .

- El caso más sencillo es clasificación binaria,  $k = 2$ .
- Ejemplos: clasificación de imágenes, reconocimiento de voz.

**Predicción estructurada:** La salida deseada es un objeto con estructura, e.g. grafo

# Formas de aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

**Regresión:** La salida deseada es un número real  $y \in \mathbb{R}$  o un vector de números reales  $\mathbf{y} \in \mathbb{R}^m$ .

- Predecir la temperatura al mediodía del jueves
- Predecir el valor de la nafta en 6 meses

**Clasificación:** La salida deseada es una variable categorica,  $y \in \{1, \dots, k\}$ .

- El caso más sencillo es clasificación binaria,  $k = 2$ .
- Ejemplos: clasificación de imágenes, reconocimiento de voz.

**Predicción estructurada:** La salida deseada es un objeto con estructura, e.g. grafo

- Estimación de posiciones humanas en imágenes

# Aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

# Aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

## Proceso de aprendizaje:

- Una familia de funciones  $f \in \mathcal{F}$  (predicción)
- Una medida discrepancia (o ajuste),  $L(f(\mathbf{x}), y)$ , entre predicciones  $f(\mathbf{x})$  y objetivos  $y$ .
- Método para elegir  $f^* \in \mathcal{F}$  de forma de:

$$f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n L(f(\mathbf{x}_i), y_i)$$

# Aprendizaje supervisado

Observamos un conjunto de pares  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$  donde  $\mathbf{x} \in \mathbb{R}^d$

## Proceso de aprendizaje:

- Una familia de funciones  $f \in \mathcal{F}$  (predicción)
- Una medida discrepancia (o ajuste),  $L(f(\mathbf{x}), y)$ , entre predicciones  $f(\mathbf{x})$  y objetivos  $y$ .
- Método para elegir  $f^* \in \mathcal{F}$  de forma de:

$$f^* = \arg \min_{f \in \mathcal{F}} \sum_{i=1}^n L(f(\mathbf{x}_i), y_i)$$

## En este curso:

- $\mathcal{F}$  es una familia paramétrica de funciones,  $f(\mathbf{x}; \theta) = f_\theta(\mathbf{x})$
- Aprendizaje se reduce a resolver el problema de optimización,

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n L(f_\theta(\mathbf{x}_i), y_i)$$

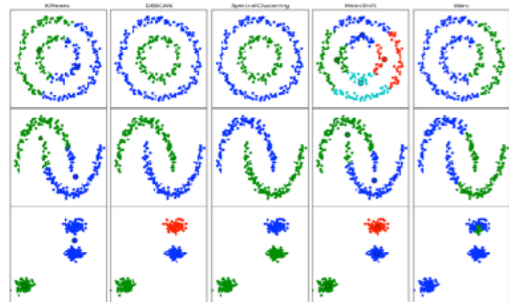
# Categorías de los sistemas de RP

## no supervisado

no se conocen las etiquetas (clases) de los datos, se organizan a partir de las características.

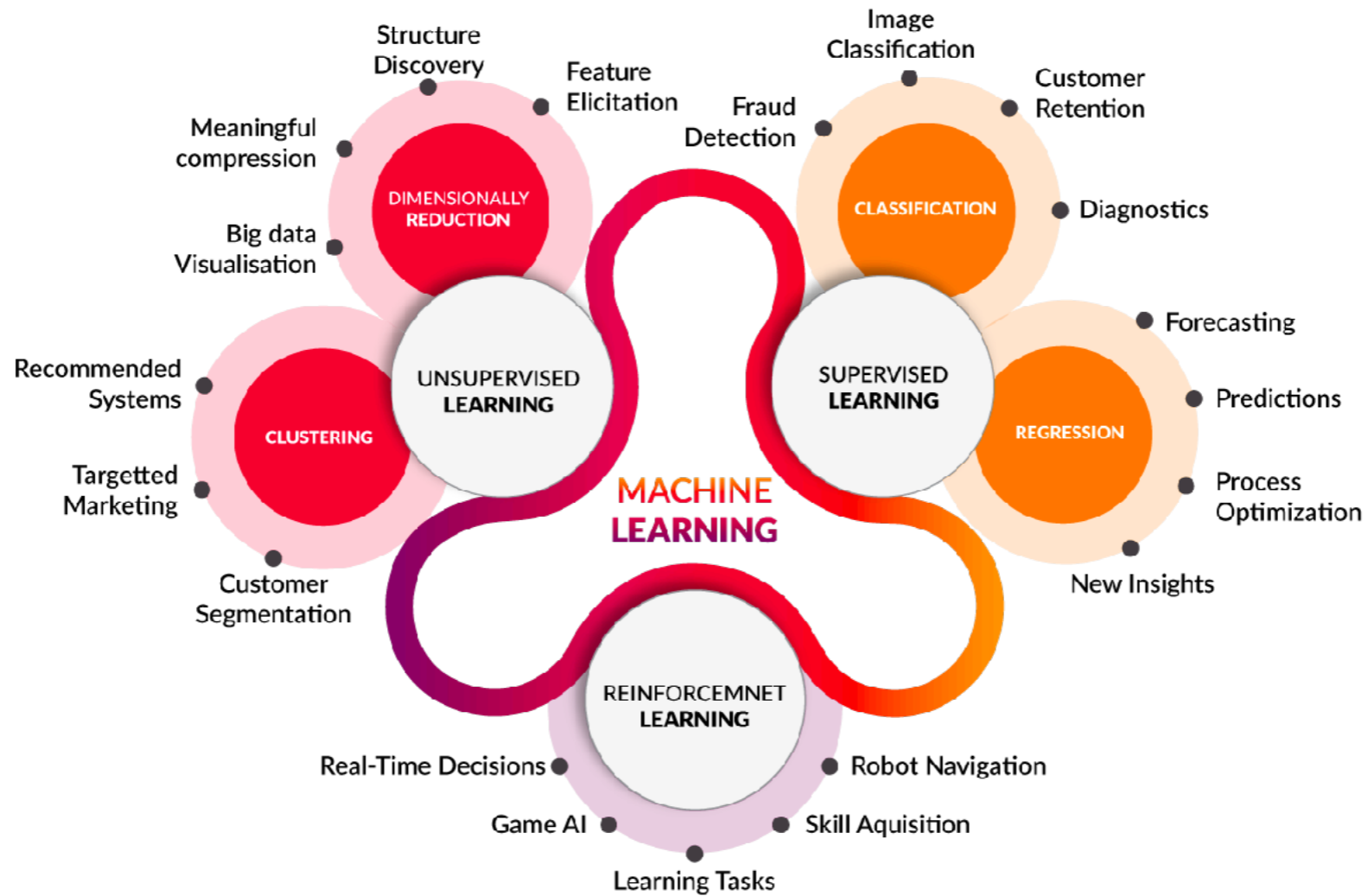
- identificación de patrones/ estructura
- evaluación indirecta o cualitativa
- organiza/agrupa clustering
- paramétrico o no
- # clases conocida o no

k-means, Fuzzy C-means, Hierarchical Clustering, Spectral Clustering, Gaussian Mixtures, Hidden Markov Model, Neural Networks, (Generalized/Robust) PCA, Isomap, MDS, Diffusion Maps, ...



## semi-supervisado

se conocen las etiquetas (clases) de algunos datos



## supervisado

datos de entrenamiento con etiquetas (clases) o valores de salida correctos para predecir datos nuevos

- clasificación y regresión
- aprendizaje explícito
- evaluación directa
- predicción clase/valor
- paramétrico o no

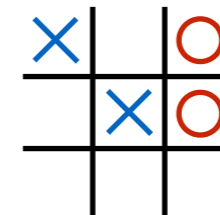
Nearest Neighbor, Support Vector Machines (SVM), Decision Trees, Random Forest, Discriminant Analysis, Naive Bayes, Neural Networks, Linear Regression, SVR, ...



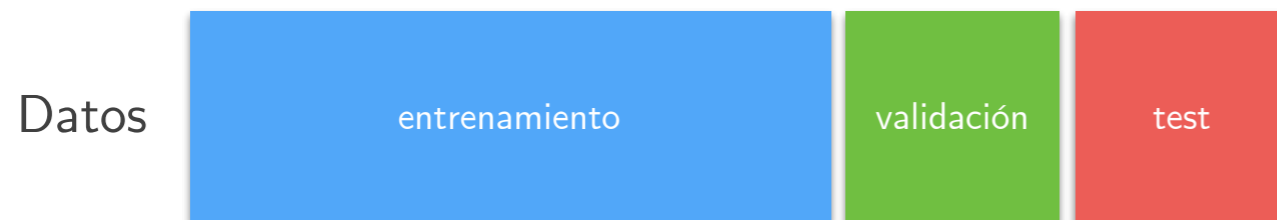
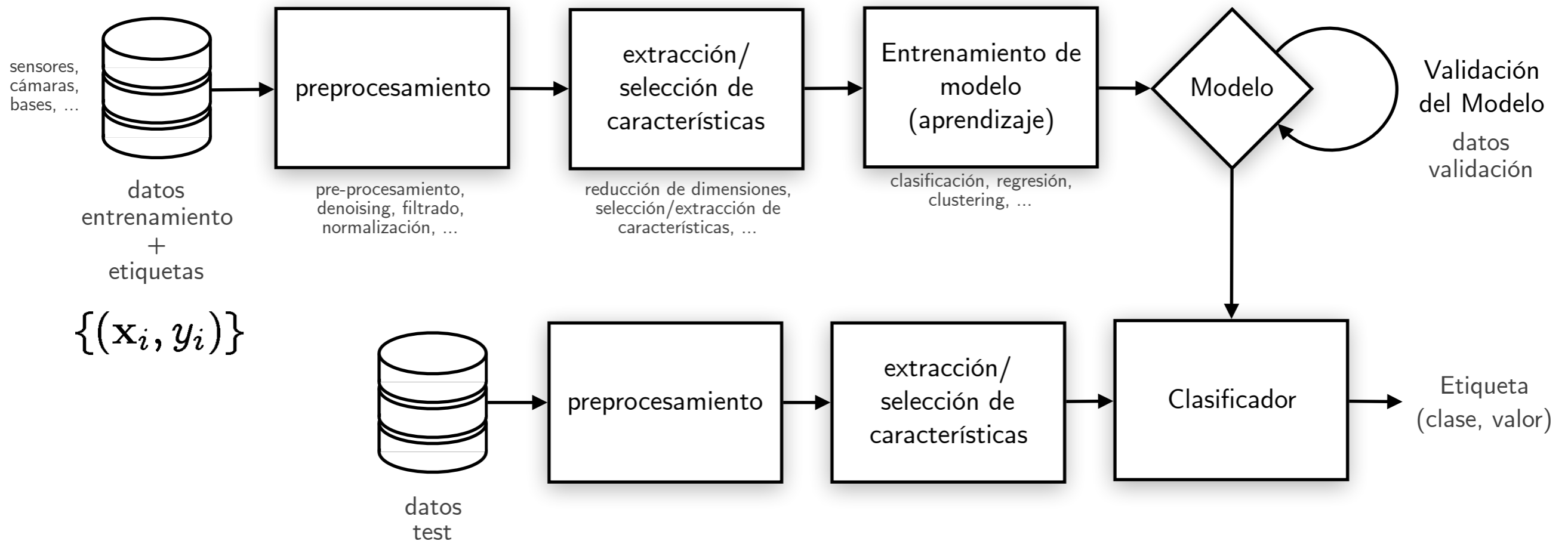
## reinforcement learning

realimentación del resultado de la tarea: recompensa o penalización

- aproximación a IA
- definir estrategias ante eventos
- maximizar recompensa

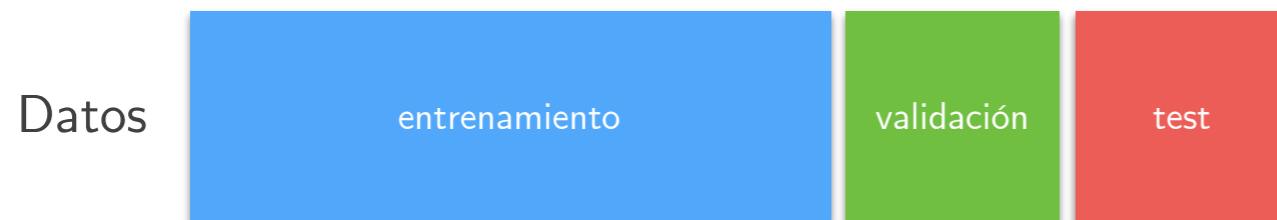
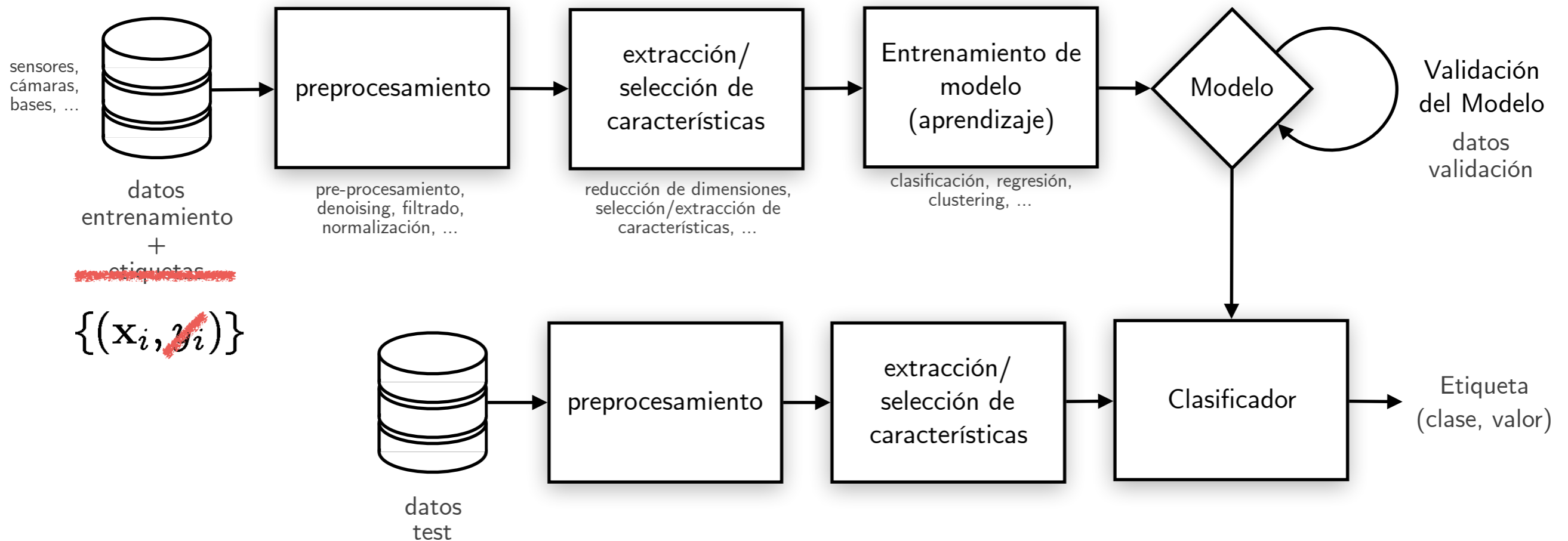


# Sistema de reconocimiento de patrones

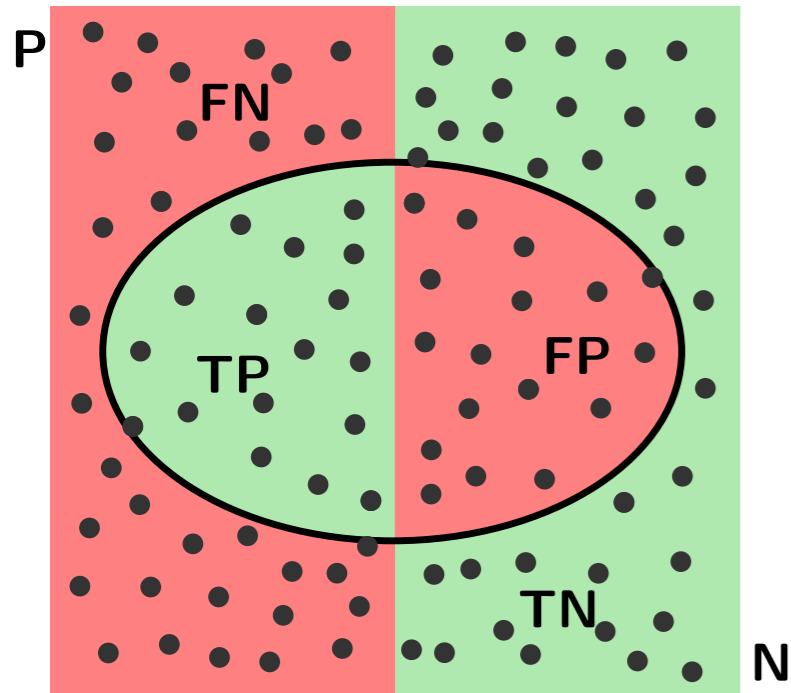




# Sistema de reconocimiento de patrones



# Medidas de desempeño



acertado                      rechazo correcto                      falsa alarma (error I)                      desacierto (error II)  
 TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

login                      enfermedad

Accuracy (exactitud): ¿cuánto se acerca a los valores reales?

Sensitivity/Recall (sensibilidad): ¿cuántos enfermos son correctamente detectados?

Specificity (especificidad): ¿cuántos sanos no son seleccionados?

Precision (presición): ¿cuántos son los enfermos de los seleccionados?

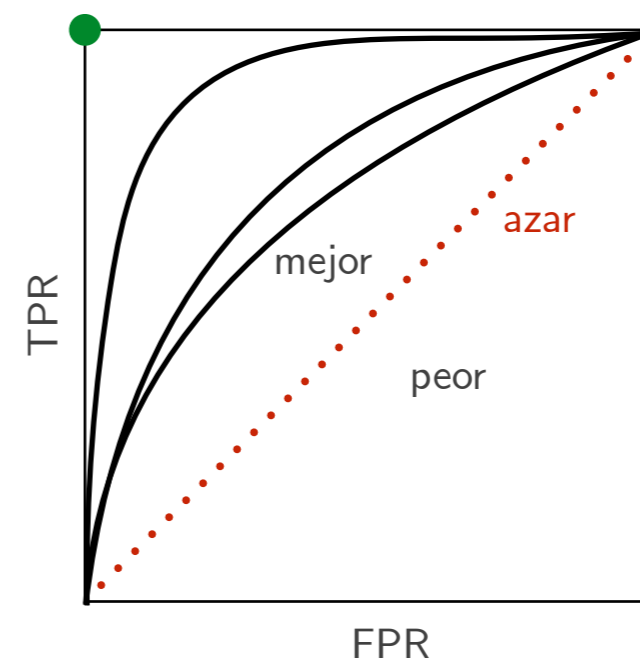
False Positive Rate: ¿cuántos sanos son seleccionados?

F1 score: Media armónica entre TPR y PPV.

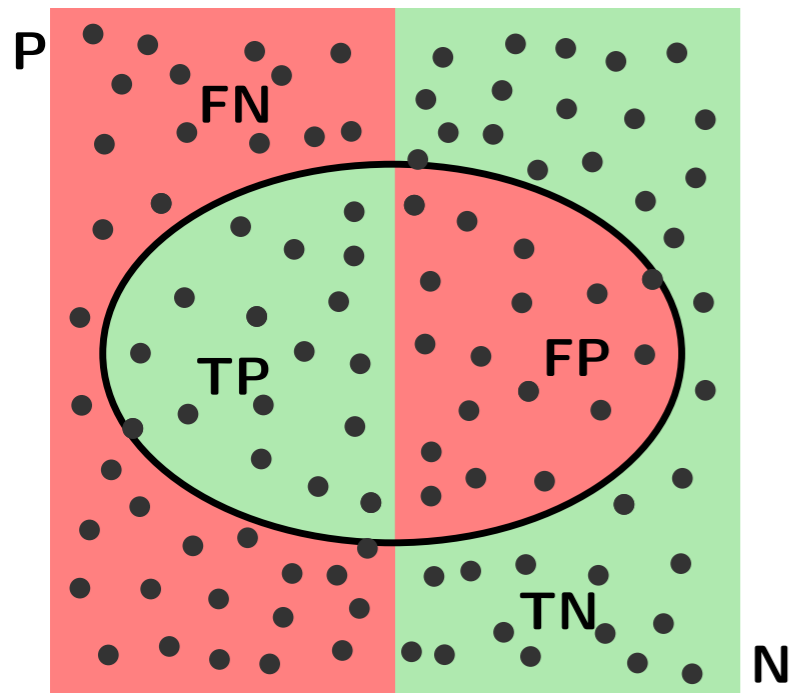
Matriz de confusión

		Valores reales	
		enfermos	sanos
Valores predichos	enfermos	TP	FP
	sanos	FN	TN

Receiver Operating Curve (ROC)



# Medidas de desempeño



acertado                      rechazo correcto                      falsa alarma (error I)                      desacierto (error II)  
 TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

login                      enfermedad

Accuracy (exactitud):  $ACC = (TP + TN) / (T + N)$

Sensitivity/Recall (sensibilidad): ¿cuántos enfermos son correctamente detectados?

Specificity (especificidad): ¿cuántos sanos no son seleccionados?

Precision (presición): ¿cuántos son los enfermos de los seleccionados?

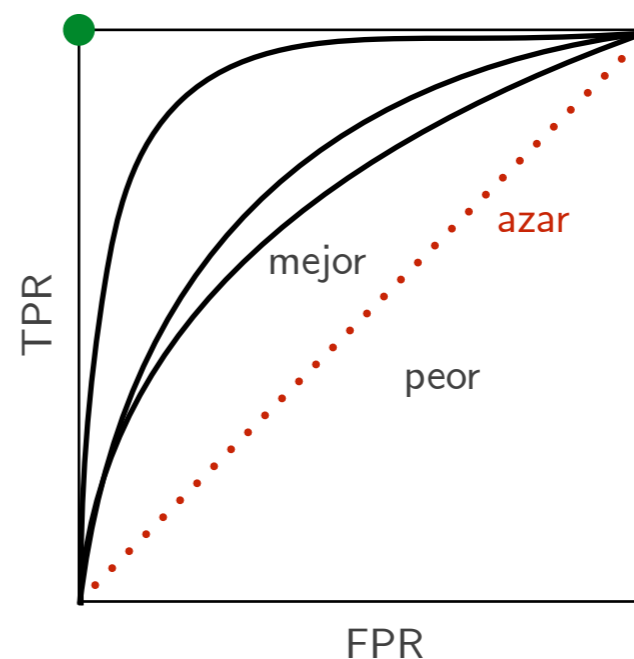
False Positive Rate: ¿cuántos sanos son seleccionados?

F1 score: Media armónica entre TPR y PPV.

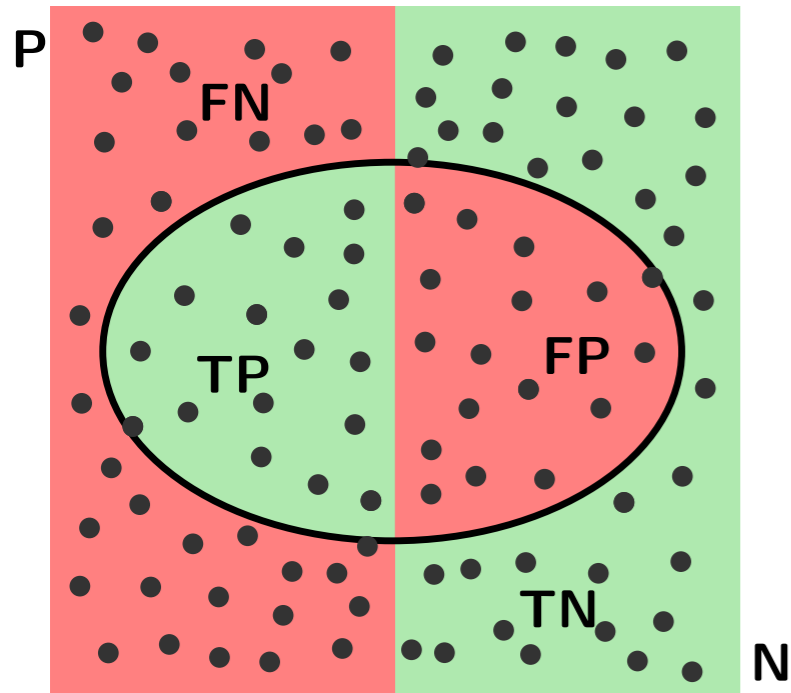
Matriz de confusión

		Valores reales	
		enfermos	sanos
Valores predichos	enfermos	TP	FP
	sanos	FN	TN

Receiver Operating Curve (ROC)



# Medidas de desempeño



acertado                      rechazo correcto                      falsa alarma (error I)                      desacierto (error II)  
 TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

login                      enfermedad

Accuracy (exactitud):  $ACC = (TP + TN)/(T + N)$

Sensitivity/Recall (sensibilidad):  $TPR = TP/P = TP/(TP + FN)$

Specificity (especificidad): ¿cuántos sanos no son seleccionados?

Precision (presición): ¿cuántos son los enfermos de los seleccionados?

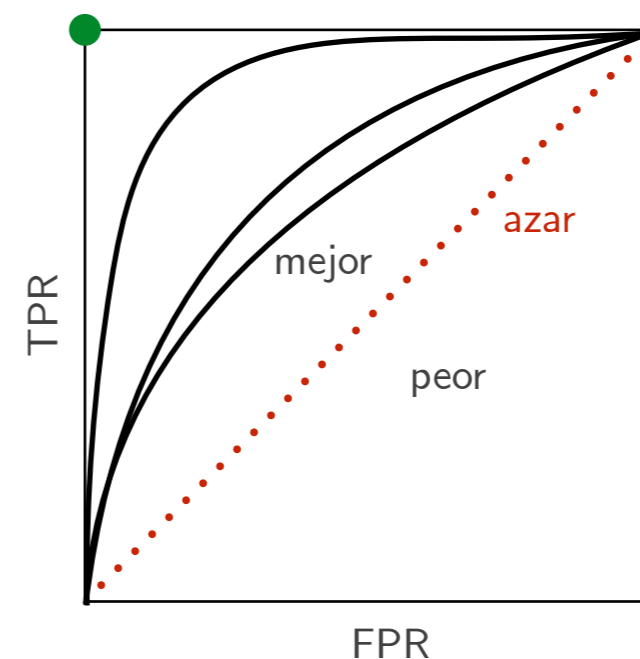
False Positive Rate: ¿cuántos sanos son seleccionados?

F1 score: Media armónica entre TPR y PPV.

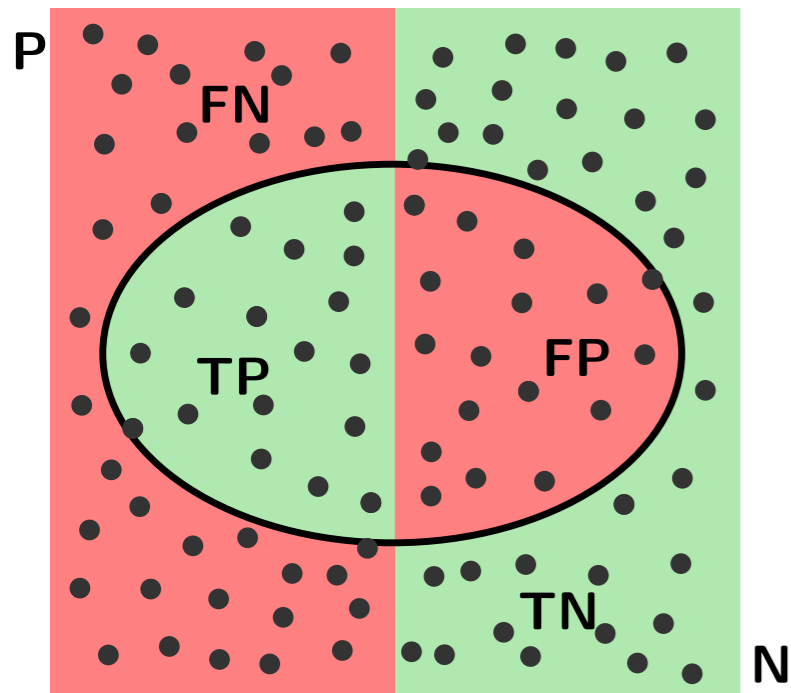
Matriz de confusión

		Valores reales	
		enfermos	sanos
Valores predichos	enfermos	TP	FP
	sanos	FN	TN

Receiver Operating Curve (ROC)



# Medidas de desempeño



acertado                      rechazo correcto                      falsa alarma (error I)                      desacierto (error II)  
 TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

login                      enfermedad

Accuracy (exactitud):  $ACC = (TP + TN)/(T + N)$

Sensitivity/Recall (sensibilidad):  $TPR = TP/P = TP/(TP + FN)$

Specificity (especificidad):  $TNR = TN/N = TN/(TN + FP)$

Precision (presición): ¿cuántos son los enfermos de los seleccionados?

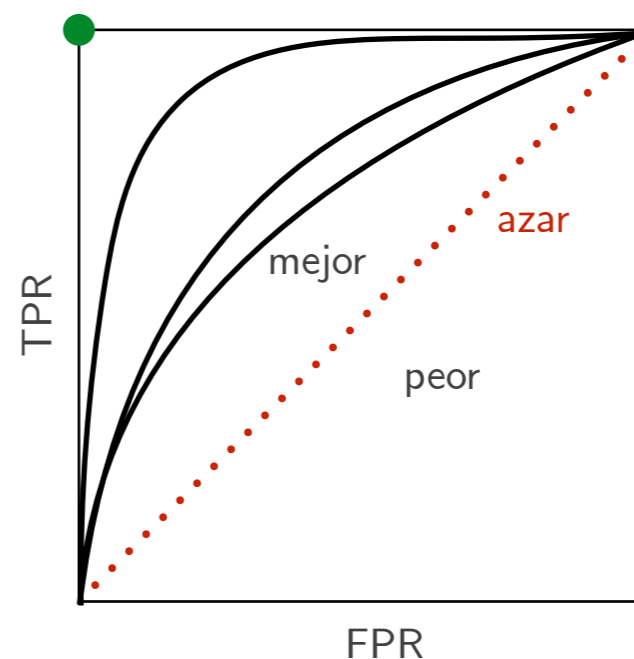
False Positive Rate: ¿cuántos sanos son seleccionados?

F1 score: Media armónica entre TPR y PPV.

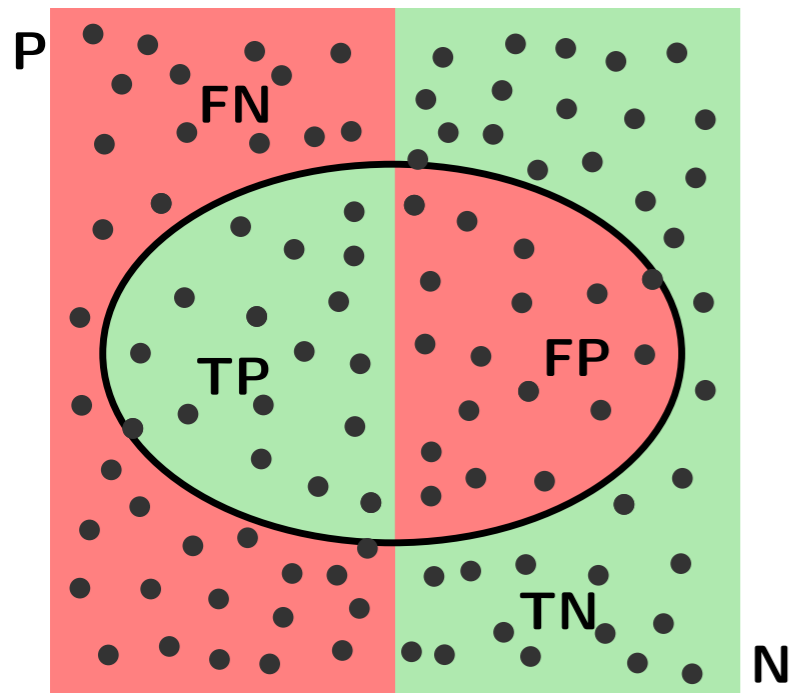
Matriz de confusión

		Valores reales	
		enfermos	sanos
Valores predichos	enfermos	TP	FP
	sanos	FN	TN

Receiver Operating Curve (ROC)



# Medidas de desempeño



acertado                      rechazo correcto                      falsa alarma (error I)                      desacierto (error II)  
 TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

login                      enfermedad

Accuracy (exactitud):  $ACC = (TP + TN)/(T + N)$

Sensitivity/Recall (sensibilidad):  $TPR = TP/P = TP/(TP + FN)$

Specificity (especificidad):  $TNR = TN/N = TN/(TN + FP)$

Precision (presición):  $PPV = TP/(TP + FP)$

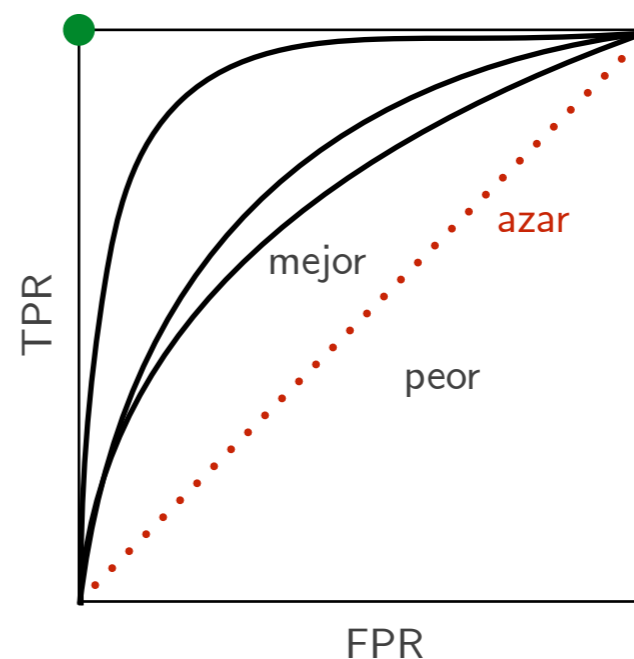
False Positive Rate: ¿cuántos sanos son seleccionados?

F1 score: Media armónica entre TPR y PPV.

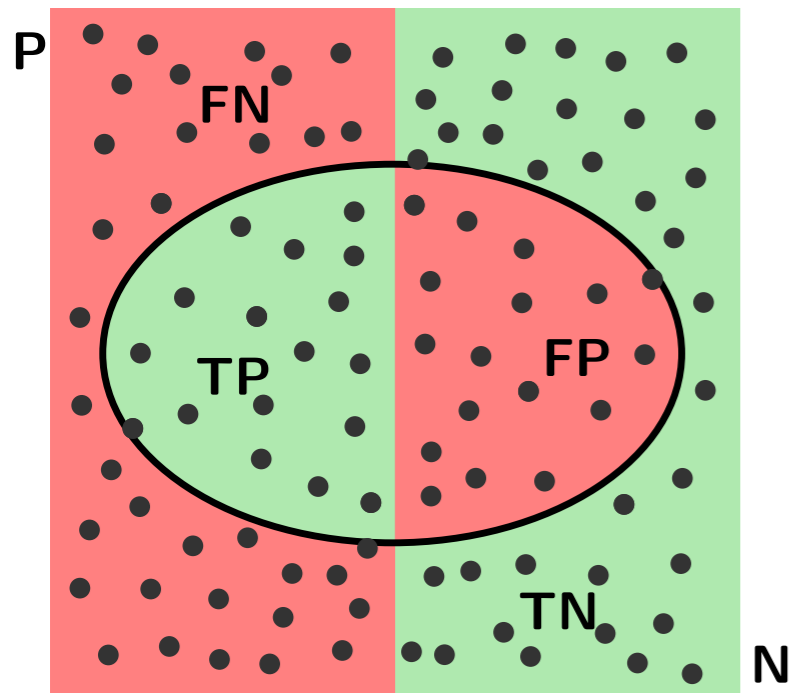
Matriz de confusión

		Valores reales	
		enfermos	sanos
Valores predichos	enfermos	TP	FP
	sanos	FN	TN

Receiver Operating Curve (ROC)



# Medidas de desempeño



acertado                      rechazo correcto                      falsa alarma (error I)                      desacierto (error II)  
 TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

login                      enfermedad

Accuracy (exactitud):  $ACC = (TP + TN)/(T + N)$

Sensitivity/Recall (sensibilidad):  $TPR = TP/P = TP/(TP + FN)$

Specificity (especificidad):  $TNR = TN/N = TN/(TN + FP)$

Precision (presición):  $PPV = TP/(TP + FP)$

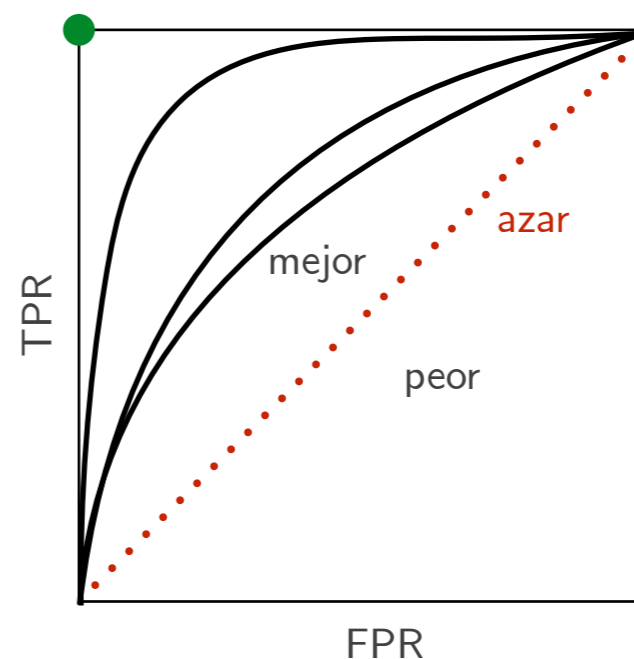
False Positive Rate:  $FPR = FP/N$

F1 score: Media armónica entre TPR y PPV.

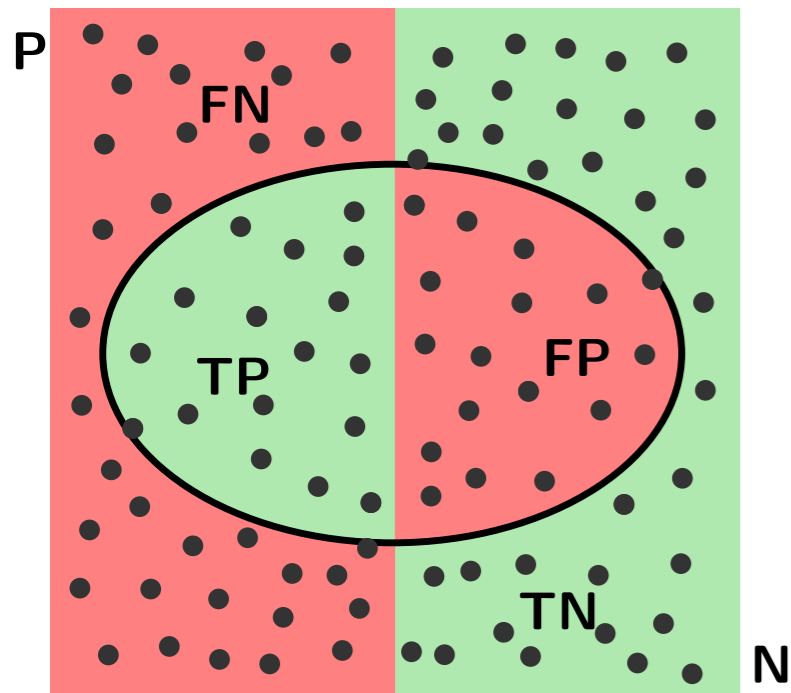
Matriz de confusión

		Valores reales	
		enfermos	sanos
Valores predichos	enfermos	TP	FP
	sanos	FN	TN

Receiver Operating Curve (ROC)



# Medidas de desempeño



acertado                      rechazo correcto                      falsa alarma (error I)                      desacierto (error II)  
 TP: true positives, TN: true negatives, FP: false positives, FN: false negatives.

login                      enfermedad

Accuracy (exactitud):  $ACC = (TP + TN)/(T + N)$

Sensitivity/Recall (sensibilidad):  $TPR = TP/P = TP/(TP + FN)$

Specificity (especificidad):  $TNR = TN/N = TN/(TN + FP)$

Precision (presición):  $PPV = TP/(TP + FP)$

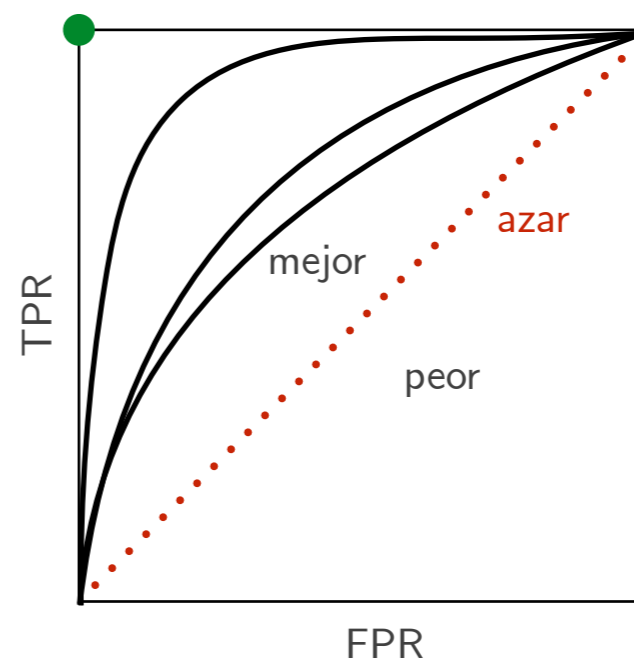
False Positive Rate:  $FPR = FP/N$

F1 score:  $F_1 = TPR \times PPV/(TPR + PPV)$

Matriz de confusión

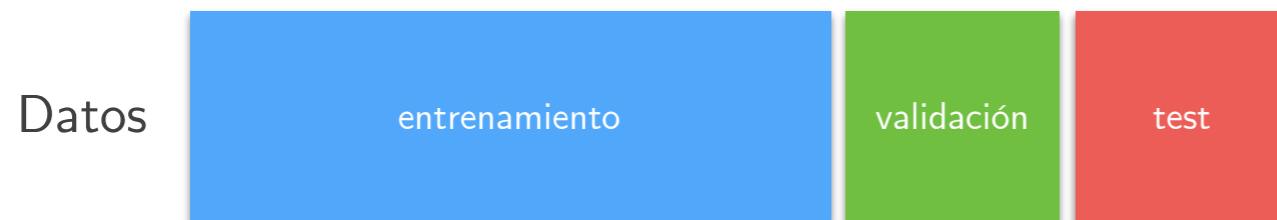
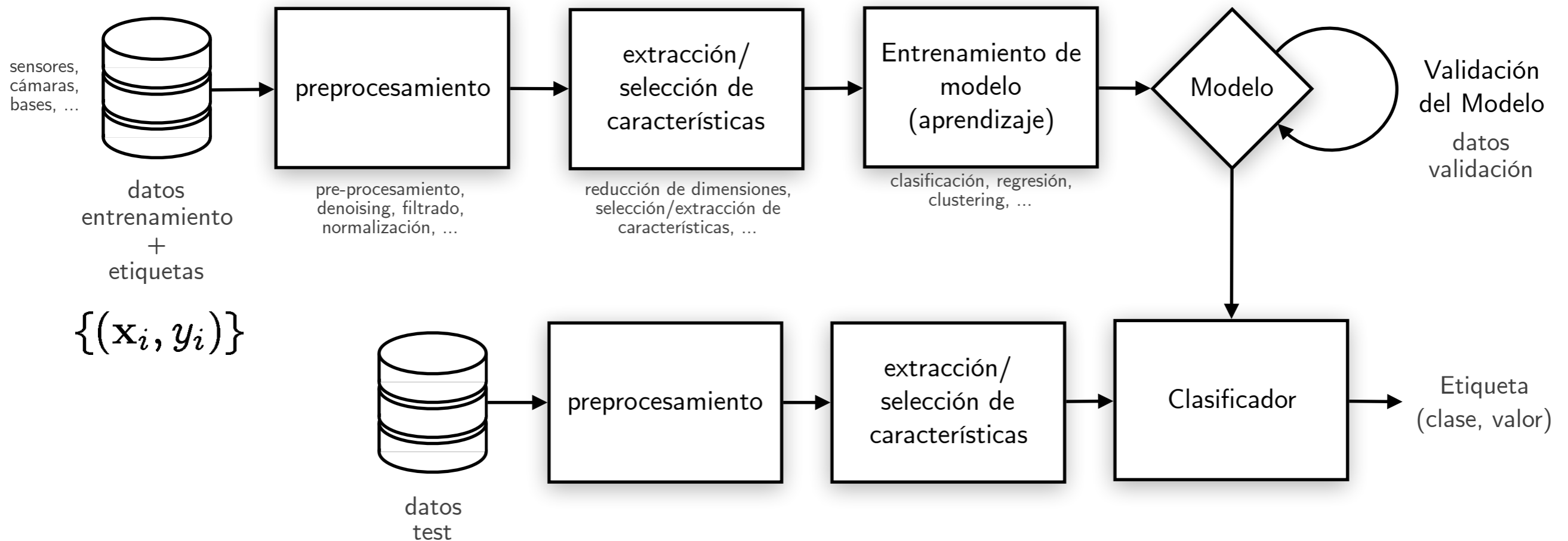
		Valores reales	
		enfermos	sanos
Valores predichos	enfermos	TP	FP
	sanos	FN	TN

Receiver Operating Curve (ROC)





# Sistema de reconocimiento de patrones



# Selección y extracción de características

- Extraer la información que puede permitir la discriminación.
- Eliminar información redundante e irrelevante.
- Reducir la dimensionalidad del problema.
- **Extracción:** crear nuevas características combinación de las características originales. Determinar un conjunto de dimensionalidad *menor* en el espacio original.
- **Selección:** seleccionar las características (originales o transformadas en el proceso de extracción) con mayor poder de discriminación.
- Diseñadas por expertos en el tema o descubiertas por su capacidad de discriminación.



Botella de *vidrio* de altura  $h_b$ , ancho de base  $w_b$ , altura pico  $h_n$  y ancho de pico  $w_n$

# Selección y extracción de características

- Extraer la información que puede permitir la discriminación.
- Eliminar información redundante e irrelevante.
- Reducir la dimensionalidad del problema.
- **Extracción:** crear nuevas características combinación de las características originales. Determinar un conjunto de dimensionalidad *menor* en el espacio original.
- **Selección:** seleccionar las características (originales o transformadas en el proceso de extracción) con mayor poder de discriminación.
- Diseñadas por expertos en el tema o descubiertas por su capacidad de discriminación.



Botella de *vidrio* de altura  $h_b$ , ancho de base  $w_b$ , altura pico  $h_n$  y ancho de pico  $w_n$



¿Descripción?

# Árboles de decisión y Bosques aleatorios

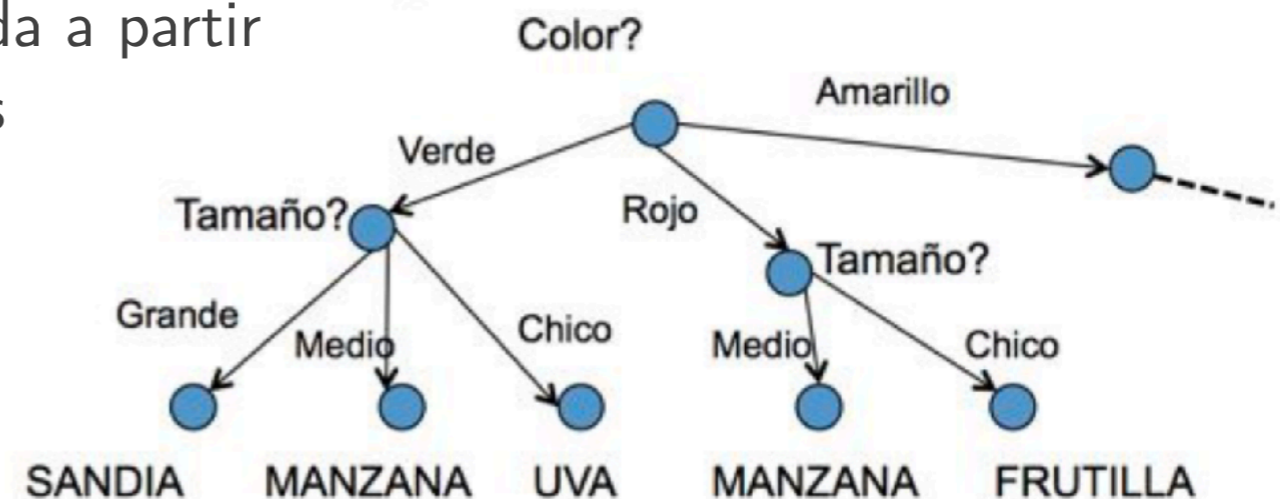
- Árboles de decisión (DTs): método no paramétrico supervisado para clasificación y regresión (CART). Busca crear un modelo para predecir el valor de salida a partir de *aprender* decisiones simples inferidas de las características.

- Ventajas

- Simple de entender, interpretar y visualizar.
- No requiere mucha preparación de los datos
- Puede manejar variables numéricas y categóricas.

- Desventajas:

- Sobreajuste (overfitting) a conjunto de entrenamiento.
- Inestables: pequeños cambios en datos dan árboles muy diferentes.
- Se aprende de heurísticas greedy que no garantizan optimalidad global
- Bias hacia una clase dominante.



color, tamaño, textura, sabor, ...

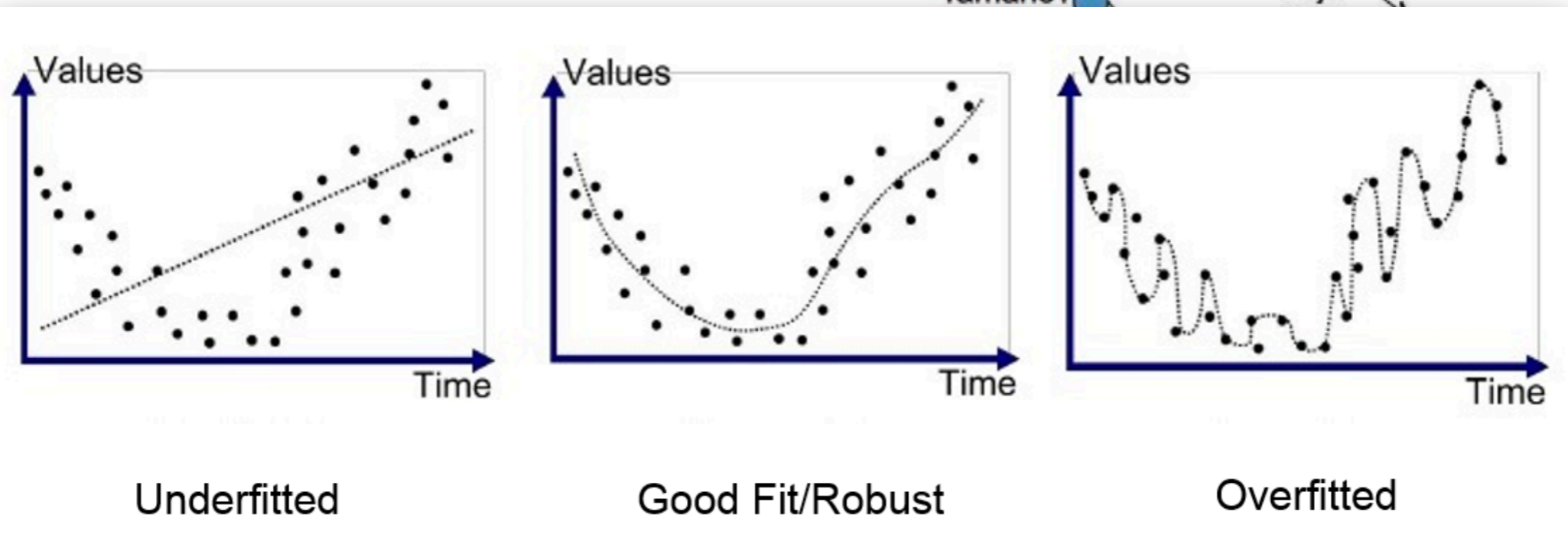
**Combinación de DTs: Random Forest**

# Árboles de decisión y Bosques aleatorios

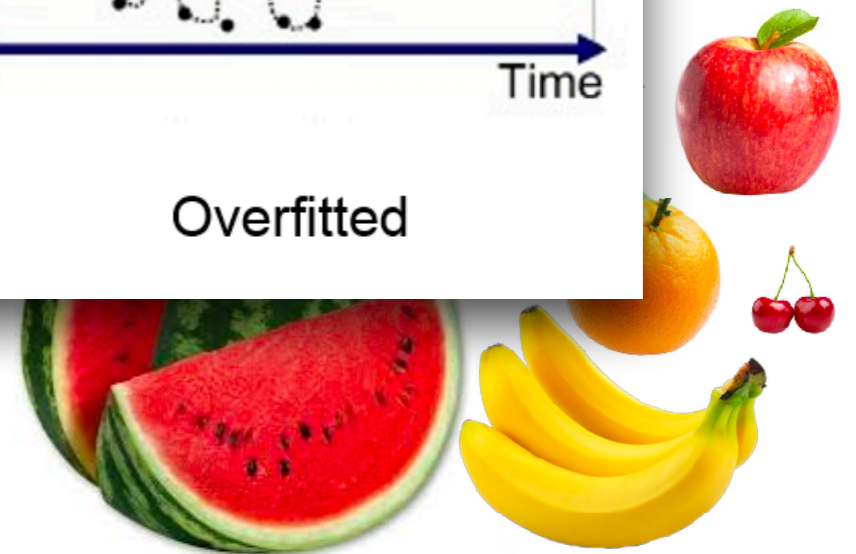
- Árboles de decisión (DTs): método no paramétrico supervisado para clasificación y regresión (CART). Busca crear un modelo para predecir el valor de salida a partir de *aprender* decisiones simples inferidas de las características.



- Ventaja
- Simple
- No requiere
- Puede
- Desventaja
- Sobre



- Inestables: pequeños cambios en datos dan árboles muy diferentes.
- Se aprende de heurísticas greedy que no garantizan optimalidad global
- Bias hacia una clase dominante.

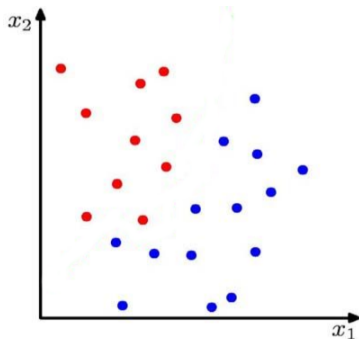


color, tamaño, textura, sabor, ...

**Combinación de DTs: Random Forest**

# Clasificación

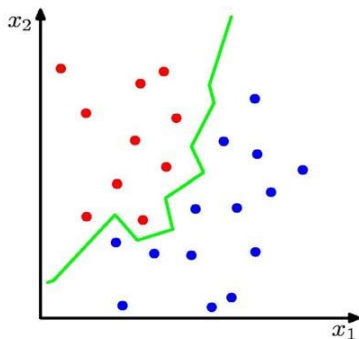
Supongamos que tenemos una base con  $N$  ejemplos (puntos de entrenamiento)  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ , tal que  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $y_i \in \{-1, 1\}$



# Clasificación

Supongamos que tenemos una base con  $N$  ejemplos (puntos de entrenamiento)  $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ , tal que  $\mathbf{x}_i \in \mathbb{R}^d$ ,  $y_i \in \{-1, 1\}$

**Objetivo:** Estimar  $f(\mathbf{x})$  de manera tal que:  $y_i \cdot f(\mathbf{x}_i) > 0, i = 1, \dots, N$ .



## Clasificador de $K$ -vecinos más cercanos ( $K$ -NN)

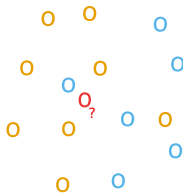
Se asigna la **clase dominante** dentro de los  **$K$**  puntos más cercanos del conjunto de entrenamiento.



# Clasificador de $K$ -vecinos más cercanos ( $K$ -NN)

Se asigna la **clase dominante** dentro de los  **$K$**  puntos más cercanos del conjunto de entrenamiento.

Dado un nuevo punto  $\mathbf{x}$  (conjunto de *test*):

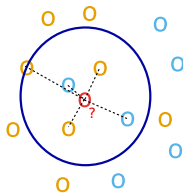


# Clasificador de $K$ -vecinos más cercanos (K-NN)

Se asigna la **clase dominante** dentro de los  **$K$**  puntos más cercanos del conjunto de entrenamiento.

Dado un nuevo punto  $\mathbf{x}$  (conjunto de *test*):

- 1 Se buscan los  **$K$**  vecinos más cercanos en el conjunto de entrenamiento

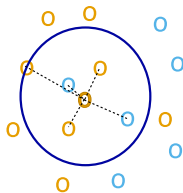


# Clasificador de $K$ - vecinos más cercanos (K-NN)

Se asigna la **clase dominante** dentro de los  **$K$**  puntos más cercanos del conjunto de entrenamiento.

Dado un nuevo punto  **$x$**  (conjunto de *test*):

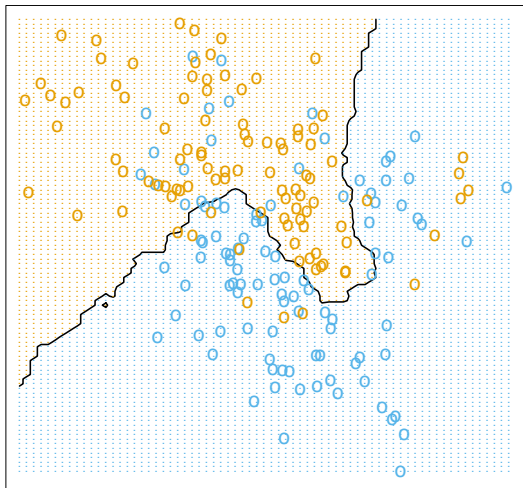
- 1 Se buscan los  **$K$**  vecinos más cercanos en el conjunto de entrenamiento
- 2 Se clasifica a  **$x$**  con la etiqueta dominante entre los  **$K$**  vecinos.



# Clasificador de $K$ -vecinos más cercanos (K-NN)

Influencia de  $K$  (cantidad de vecinos)

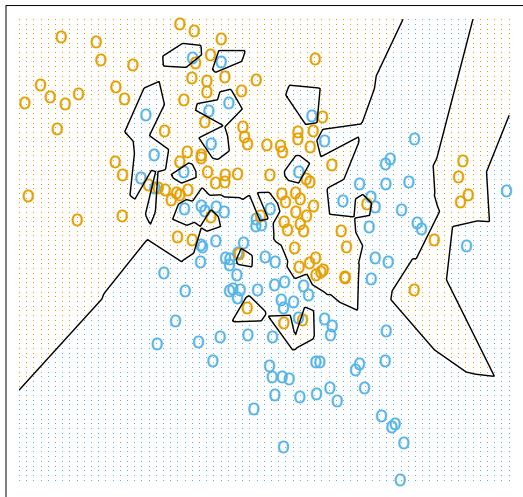
15-nn



# Clasificador de $K$ -vecinos más cercanos (K-NN)

Influencia de  $K$  (cantidad de vecinos)

1-nn

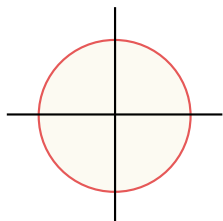


# Clasificador de $K$ -vecinos más cercanos (K-NN)

- ¿Qué función de **distancia** utilizamos? Sea  $u, v \in \mathbb{R}^d$

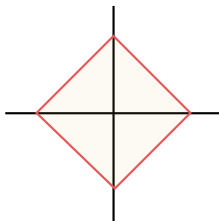
$L_2$  (Euclídea)

$$d_2(u, v) = \sqrt{\sum_i (u(i) - v(j))^2}$$



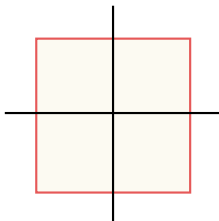
$L_1$  (Manhattan)

$$d_1(u, v) = \sum_i |u(i) - v(j)|$$



$L_\infty$  (Max)

$$d_\infty(u, v) = \max_i |u(i) - v(i)|$$



# Clasificador de $K$ -vecinos más cercanos (K-NN)

- ¿Cómo elegimos  $K$ ?
- ¿Qué función de **distancia** utilizamos?

Estos son **hiperparámetros**: los debemos elegir más que aprender (aunque no siempre es clara la diferencia).

# Clasificador de $K$ -vecinos más cercanos (K-NN)

- ¿Cómo elegimos  $K$ ?
- ¿Qué función de **distancia** utilizamos?

Estos son **hiperparámetros**: los debemos elegir más que aprender (aunque no siempre es clara la diferencia).

- La elección depende del problema.



# Clasificador de $K$ -vecinos más cercanos (K-NN)

- ¿Cómo elegimos  $K$ ?
- ¿Qué función de **distancia** utilizamos?

Estos son **hiperparámetros**: los debemos elegir más que aprender (aunque no siempre es clara la diferencia).

- La elección depende del problema.
- Hay que probar y evaluar qué funciona mejor

# Clasificador de $K$ -vecinos más cercanos (K-NN)

- ¿Cómo elegimos  $K$ ?
- ¿Qué función de **distancia** utilizamos?

Estos son **hiperparámetros**: los debemos elegir más que aprender (aunque no siempre es clara la diferencia).

- La elección depende del problema.
- Hay que probar y evaluar qué funciona mejor
- Hay que definir un protocolo de evaluación **realista**

¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso *real*?

¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

- 1 Elijo los parámetros/hiperparámetros (e.g., **K**) que minimizan el error en el conjunto de datos etiquetados que dispongo.

*Dataset*

¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

- 1 Elijo los parámetros/hiperparámetros (e.g.,  $\mathbf{K}$ ) que minimizan el error en el conjunto de datos etiquetados que dispongo.



*Dataset*

- **Mala idea...** (sobreajuste - *overfitting*).  $K = 1$  es siempre lo óptimo.

¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

2 Particiono el conjunto de datos etiquetados: **Train** + **Test**

Elijo hiperparámetros que funcionan mejor en el conjunto **Test**



¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

2 Particiono el conjunto de datos etiquetados: **Train** + **Test**

Elijo hiperparámetros que funcionan mejor en el conjunto **Test**



- **Mala idea...** No sabemos cómo será el funcionamiento en datos no conocidos (escenario real)

¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

- 3 Particiono el conjunto de datos en: **Train** + **Validation** + **Test**  
Elijo los hiperparámetros que funcionan mejor en **Validation**





¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

3 Particiono el conjunto de datos en: **Train** + **Validation** + **Test**

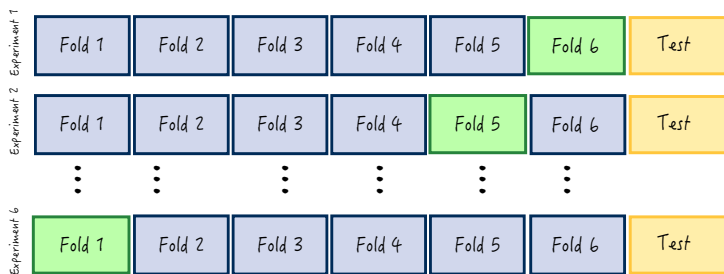
Elijo los hiperparámetros que funcionan mejor en **Validation**

Evalúo al final el desempeño en el conjunto **Test**.



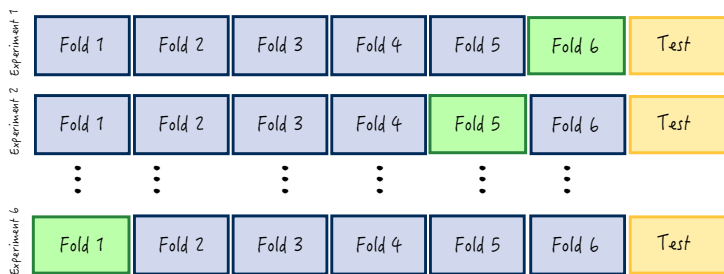
## ¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

- 4 Particiono el conjunto de datos en  $N$  subconjuntos (**folders**) + **Test**. Se toma cada subconjunto como validación y el resto para entrenamiento ( $N$  experimentos). Se promedian los resultados.



## ¿Cómo elijo los parámetros para **maximizar** y **evaluar** el **desempeño** del sistema en un escenario de uso **real**?

- 4 Particiono el conjunto de datos en  $N$  subconjuntos (**folds**) + **Test**. Se toma cada subconjunto como validación y el resto para entrenamiento ( $N$  experimentos). Se promedian los resultados.

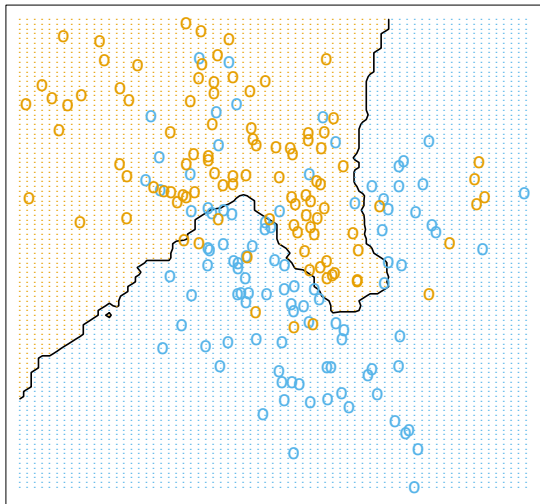


- **Validación Cruzada** es útil para conjuntos de datos pequeños, poco usado en Deep Learning (*Big data*)

# Clasificador de K-vecinos más cercanos (K-NN)

Influencia de  $K$  (cantidad de vecinos)

15-nn





# Clasificación en Python

# scikit-learn

- <https://scikit-learn.org/stable/>
- Pasos básicos de clasificación supervisada:
  - **Initialization:** Cada modelo tiene su propio constructor ej:  
`model= KneighborsClassifier(n_neighbors=5)`
  - **Training:** Cada modelo implementa un método `fit()` que entrena el modelo.  
  
ej: `model.fit(X_train, y_train)`
  - **Prediction:** Cada modelo implementa un método `predict()` que predice la salida para las nuevas muestras,  
ej `y_pred=model.predict(X_test)`
  - **Probabilities:** Muchos modelos también implementan un método `predict_proba`  
`p=model.predict_proba(X_test)`
- <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

# Vecinos más cercanos

- Parámetros:

**n\_neighbors:** K

**metric:** métrica de cálculo de la distancia.

**algorithm:** cuál es el algoritmo aplicado para encontrar el vecino más cercano (K-D Tree o fuerza bruta).

```
# Training code:  
from sklearn.neighbors import KNeighborsClassifier  
  
model = KNeighborsClassifier(n_neighbors = 5, metric = "euclidean")  
model.fit(X, y)
```

```
# Testing code:  
>>> model.predict([-10,-2])  
array([ 0.])  
  
>>> model.predict([1,2])  
array([ 1.])  
  
>>> model.predict_proba([0,-3])  
array([[ 0.6,  0.4]])  
  
# Ask what are the 5 nearest neighbors.  
# Returns the distances and indices (rows) in X  
>>> distances, indices = model.kneighbors([0,-3])  
[0.144, 0.254, 0.362, 0.414, 0.422]  
[379, 11, 215, 370, 198]  
  
# What are the classes for these five:  
>>> y[indices]  
array([[ 0.,  1.,  0.,  0.,  1.]])
```

# Árbol de decisión

- Dividir los datos en conjunto de entrenamiento y conjunto de test

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.4, random_state=213)
```

- Definimos el objeto clasificador y lo entrenamos

```
clf = tree.DecisionTreeClassifier(min_samples_leaf=2)  
clf = clf.fit(X_train, y_train)
```

- Visualizamos el grafo generado en el entrenamiento

```
import pydotplus  
from IPython.display import Image, display  
  
dot_data = tree.export_graphviz(clf, out_file=None,  
                                feature_names=feat_names,  
                                class_names=classes,  
                                filled=True, rounded=True,  
                                special_characters=True)  
  
graph = pydotplus.graph_from_dot_data(dot_data)  
graph.write_png('tree.png')  
  
display(Image('tree.png'))
```

- Predecimos las clases del conjunto de test con el modelo entrenado

```
y_pred = clf.predict(X_test)  
y_pred_train = clf.predict(X_train)
```

