INTELIGENCIA ARTIFICIAL (1INF24)

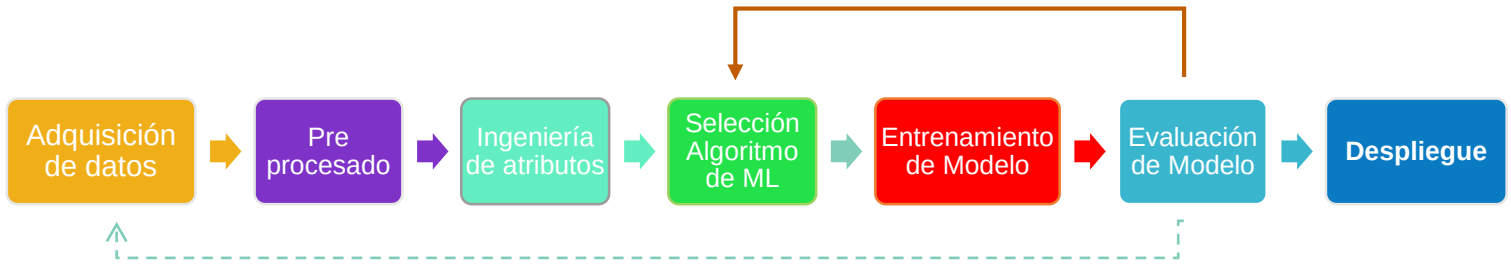
Unidad 2: Fundamentos de *Machine Learning* y redes neuronales artificiales

Tema 4: *Algoritmos Básicos para Clasificación*

Dr. Edwin Villanueva Talavera

Contenido

- El problema de clasificación
- Métricas de clasificación
- Clasificación K-NN
- Arboles de Clasificación



El Problema de Clasificación

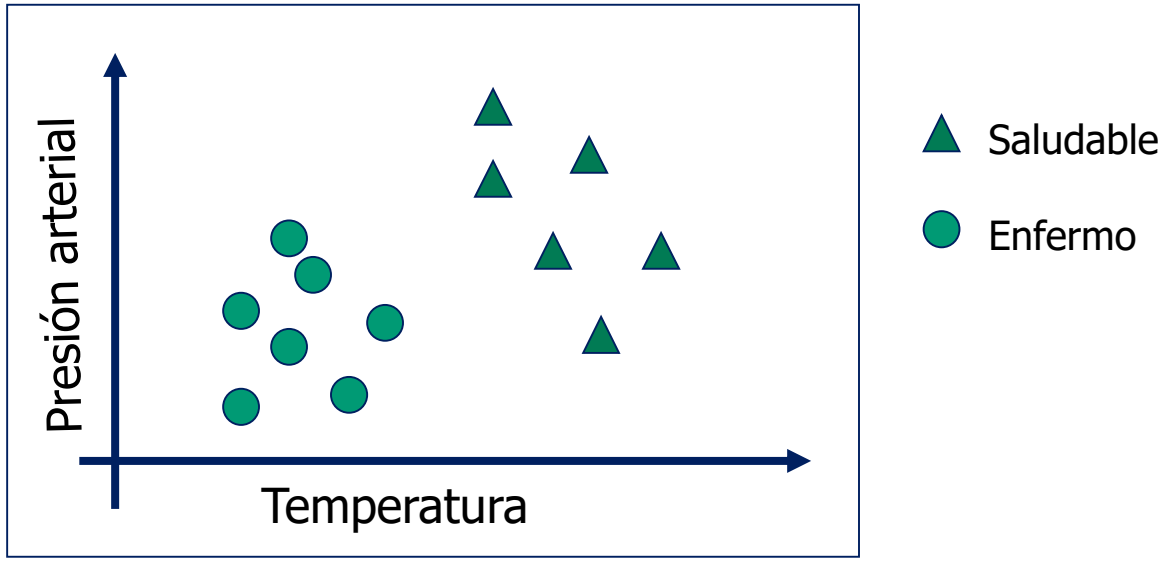
El conjunto de datos para clasificación

Variables de entrada (variables preditivas)

	Nom.	Temp.	Edad	Peso	Altura	Clase (target)
Registros (ejemplos)	Juan	37	70	94	190	Saludable
	Maria	38	65	60	172	Enfermo
	José	39	19	70	185	Enfermo
	Silvia	38	25	65	160	Saludable
	Pedro	37	70	90	168	Enfermo

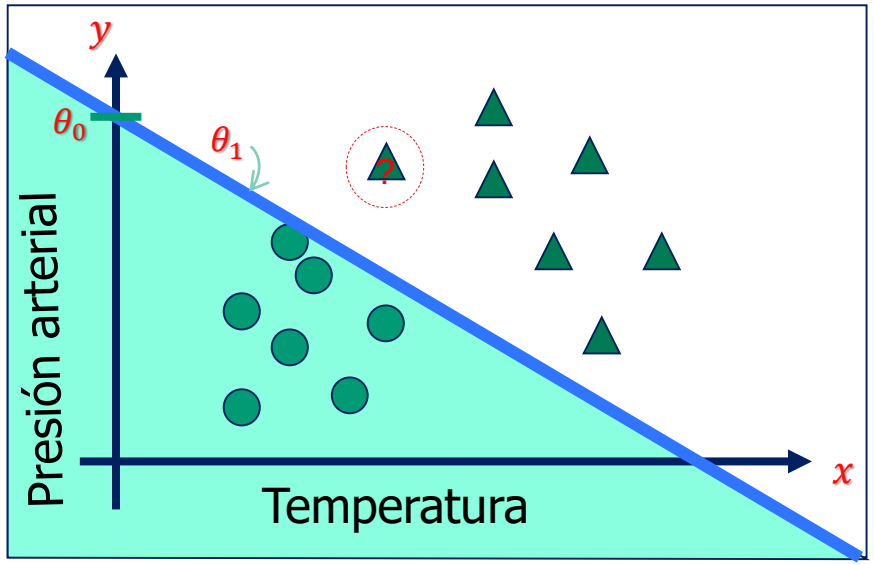
El Problema de Clasificación

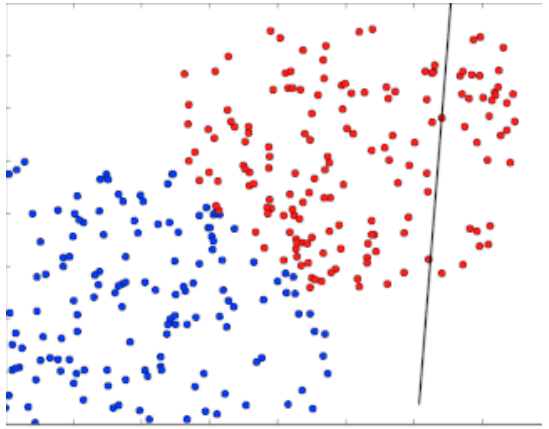
¿Cómo clasificar?



El Problema de Clasificación

Clasificador lineal





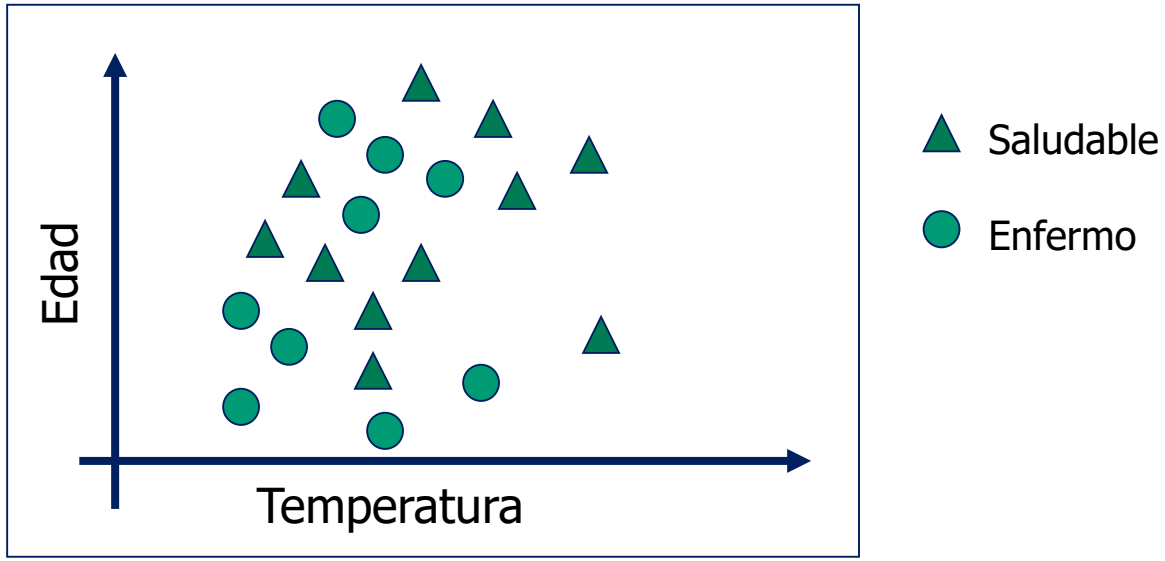
Saludable

Enfermo

$f(x) = \theta_0 + \theta_1 x$

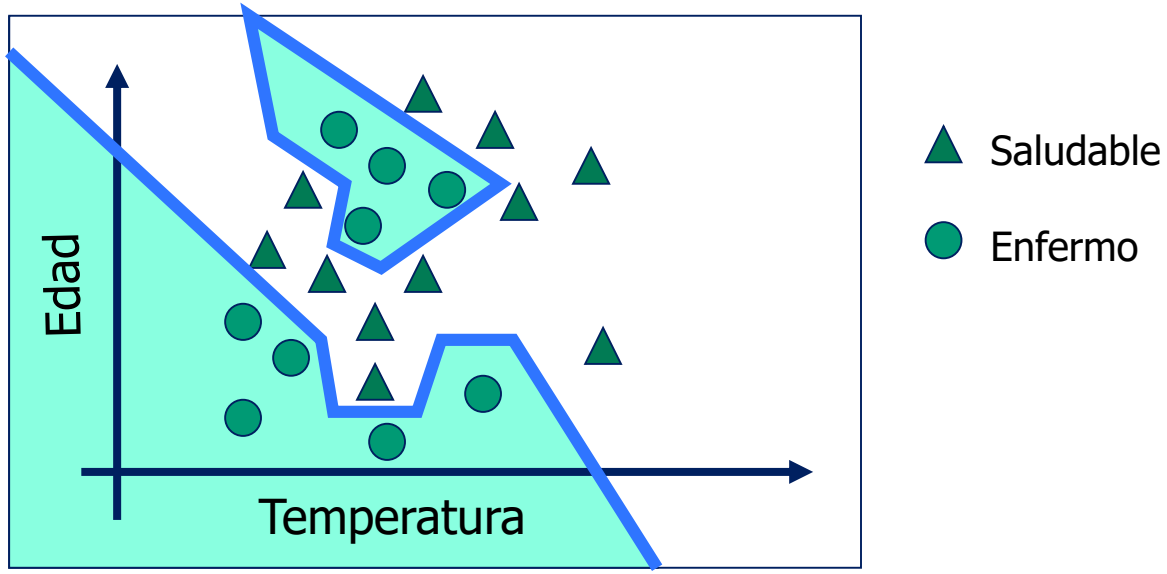
El Problema de Clasificación

¿Cómo clasificar?



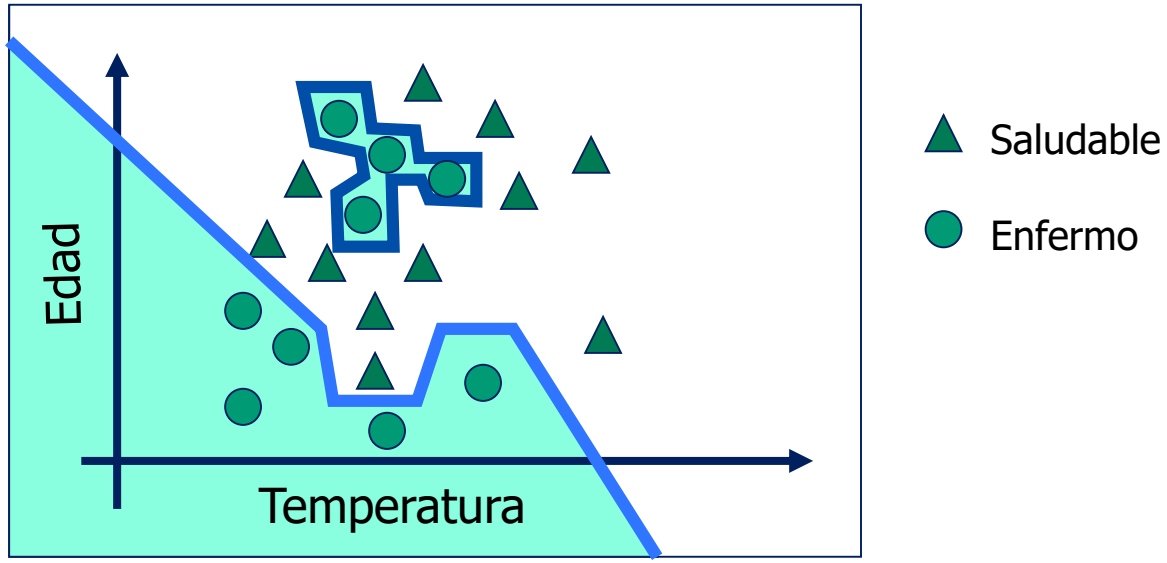
El Problema de Clasificación

Clasificación no lineal



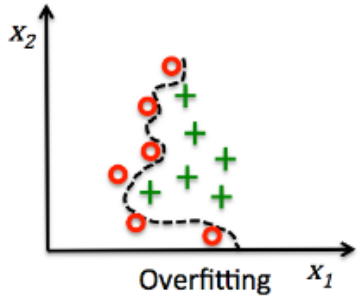
El Problema de Clasificación

Riesgo: sobre ajuste (Overfitting)

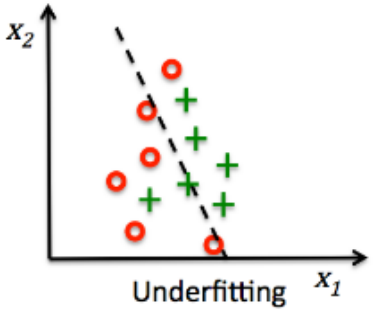


El Problema de Clasificación

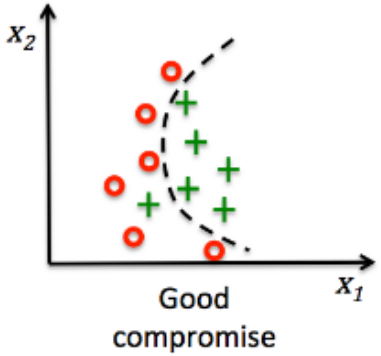
- El modelo falla al reconocer un nuevo dato porque no tiene valores similares a las muestras de entrenamiento.



- El modelo no genera la salida deseada frente a un nuevo dato por falta de ajuste del modelo. No puede generalizar.



Regularización



El Problema de Clasificación

Matriz de confusión

La matriz de confusión es una matriz de tamaño $m \times m$

m indica el número de clases

C_{ij} indica el número de instancias de la clase real i que fueron etiquetados (predichos) como clase j por el modelo.

Clase predicha

Clase real		$Clase_1$	$Clase_2$	...	$Clase_m$
	$Clase_1$	$C_{1,1}$	$C_{1,2}$	…	$C_{1,m}$
	$Clase_2$	$C_{2,1}$	$C_{2,2}$	…	$C_{2,m}$
	...	…	…	…	…
	$Clase_m$	$C_{m,1}$	$C_{m,2}$	…	$C_{m,m}$

Idealmente, la mayoría de las instancias deben estar representadas a lo largo de la diagonal de la matriz de confusión.

Métricas de clasificación

Matriz de confusión

		Clase predicha $\hat{y}$: No	Clase predicha $\hat{y}$: Si
Clase real y	No	Verdaderos Negativos **(TN)**	Falsos Positivos **(FP)**
	Si	Falsos Negativos **(FN)**	Verdaderos Positivos **(TP)**

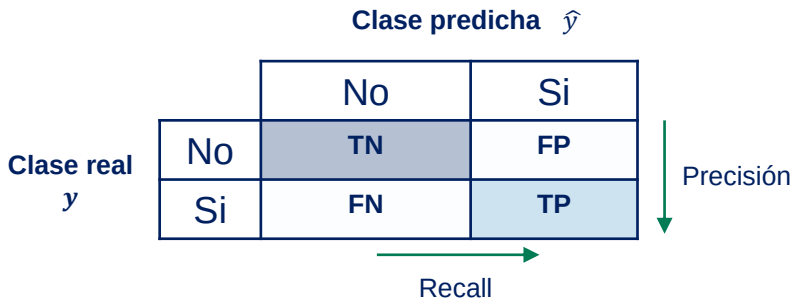
Verdaderos negativos (TN): negativos correctamente etiquetados como negativos

Verdaderos positivos (TP): positivos correctamente etiquetados como positivos

Falsos negativos (FN): positivos incorrectamente etiquetados como negativos

Falsos positivos (FP): negativos incorrectamente etiquetados como positivos

Métricas de clasificación



- acc (Accuracy) $acc = \frac{TN + TP}{TN + TP + FP + FN}$
- P (Precision) $\mathrm{P} = \frac{TP}{FP + TP}$
- R (Recall) $\mathrm{R} = \frac{TP}{FN + TP}$
- *F*1 (F1-score) $F1 = \frac{2\,(\mathrm{R} \times \mathrm{P})}{\mathrm{R} + \mathrm{P}}$
- Balanced-accuracy $0.5\,(\frac{TN}{TN+FP} + \frac{TP}{FN+TP})$

Ejemplo:

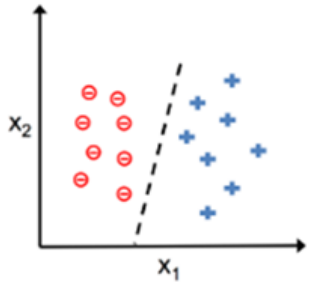
		Diagnosticada No enferma (sana) - (0)	Diagnosticada Enferma + (1)
		No	Si
Sano - (0)	No	79 (TN)	1 (FP)
Enferma + (1)	Si	12 (FN)	28 (TP)

Calcular las métricas:

acc = ?
P = ?
R = ?
F1 = ?
Balanced-accuracy = ?

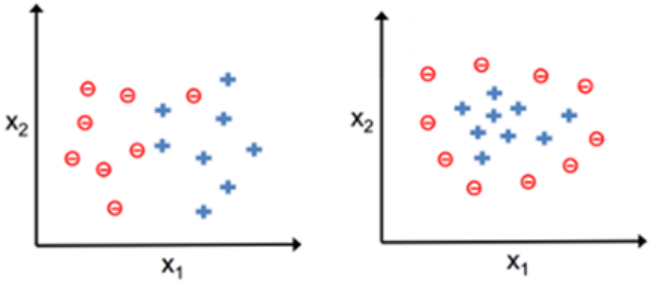
Algoritmos de Clasificación

Lineales



- Logistic Regression

No lineales

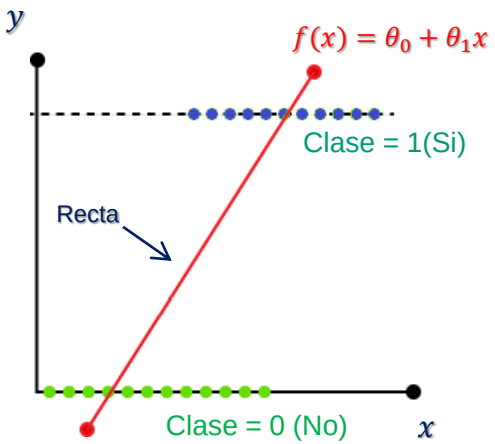


- k-Nearest Neighbors (KNN).
- Decision Trees.
- Support Vector Machines
- Redes Neuronales, etc.

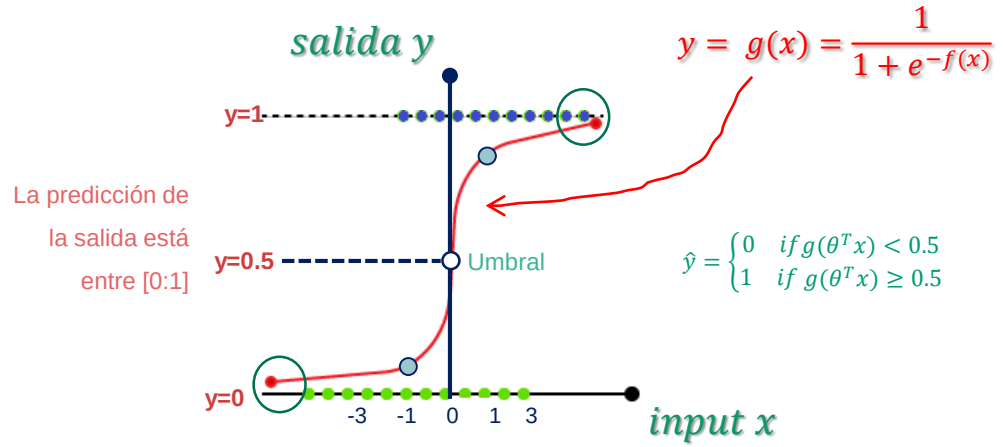
Regresión Logística

Caso univariado

Regresión Lineal

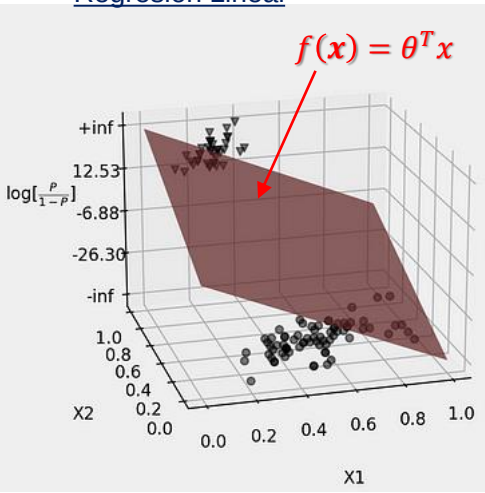


Regresión Logística

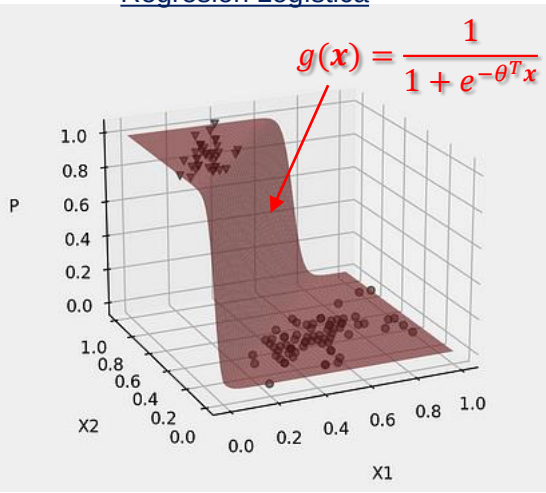


Caso mulivariado

Regresión Lineal



Regresión Logística



Notación general:

$$\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

- θ es un vector de parámetros $[\theta_0, \theta_1, \theta_2, \ldots, \theta_n]$.
- x es un vector de características $[1, x_1, x_2, \ldots, x_n]$

Función de perdida (entropía cruzada):

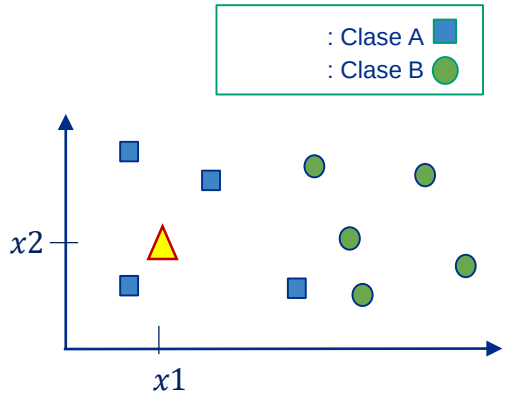
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

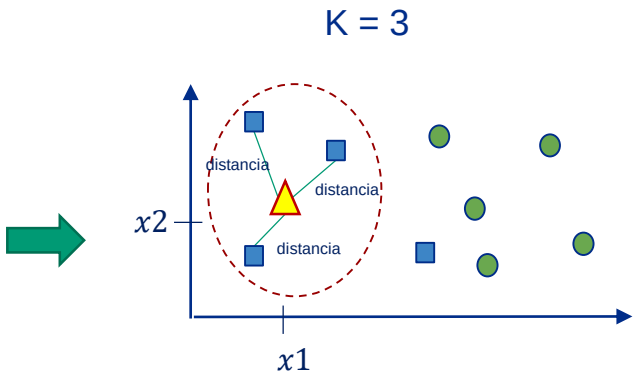
Optimización (gradiente descendente):

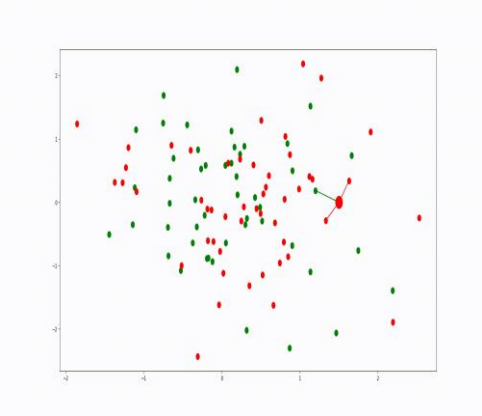
$$\theta_j := \theta_j - \alpha\frac{\partial J(\theta)}{\partial \theta_j}$$

Clasificación KNN

- KNN (k-Nearest Neighbors) es un algoritmo usado para problemas de clasificación y regresión.
- Se debe seleccionar un valor K, el cual indica la cantidad de vecinos cercanos que se consideran para etiquetar un nuevo dato.
- Para conocer cuales son los vecinos más cercanos se usará una función de distancia.



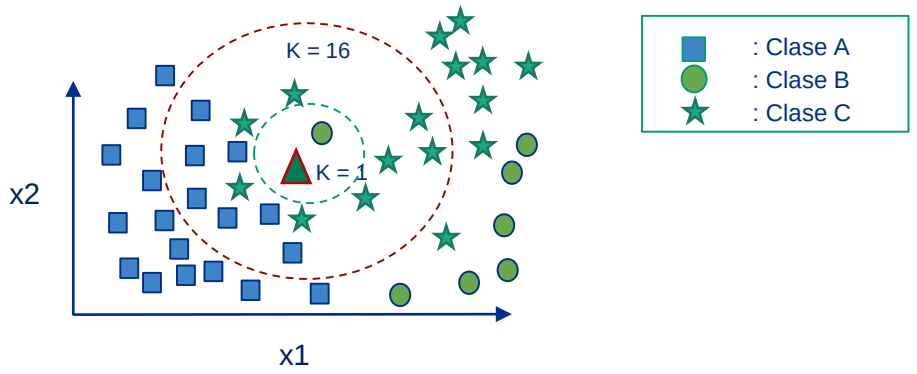


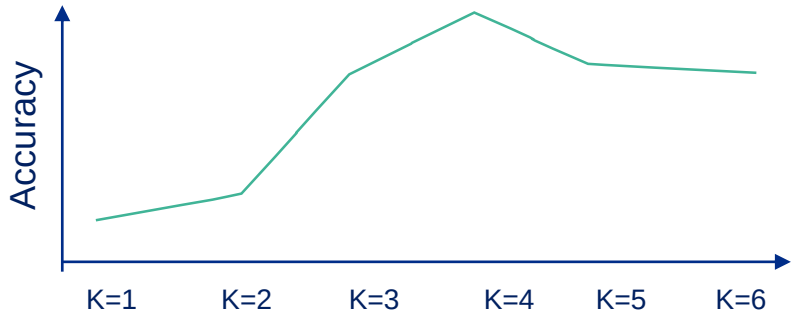


Para determinar a qué clase pertenecerá un nuevo registro, se consideran las distancias de los k vecinos adyacentes más cercanos.

Distancia de Minkowski: $d(A,B) = \sqrt[p]{\sum_{i=1}^{n}(|A(x_i) - B(x_i)|)^p}$, $\quad p \geq 1$

Clasificación KNN





Arboles de clasificación

- Algoritmo de entrenamiento (construccion del arbol)

1) Inicio: Comienza colocando todos los datos en un nodo (nodo raíz)

2) Evaluar divisiones: Para cada nodo hoja con datos D, el algoritmo evalúa cada variable y posible división (D1, D2) en cuanto a su **ganancia_de_pureza (D, D1, D2)**

3) Escoge la mejor división: Elige el nodo y la mejor división (la variable y división que maximiza la ganancia en pureza) y crear 2 nodos hijos con los datos de dicha división

4) Repetir: Repetir pasos 2 y 3 hasta que se cumpla una condición de parada (se llega a un mínimo de datos en los nodos; profundidad máxima del árbol, etc.)

ganancia_de_pureza (D, D1, D2):

$$= Impureza(D) - (w1 * Impureza(D1) + w2 * Impureza(D2))$$

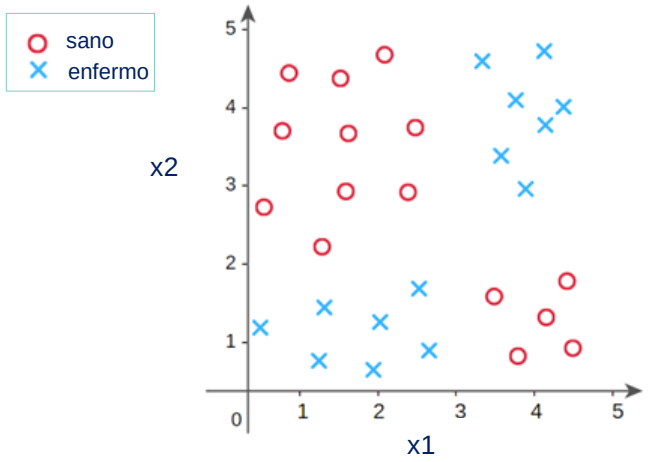
donde:
w1 y w2 son las proporciones de datos correspondientes a D1 y D2

- Prediccion en nuevos datos

Para predecir el valor de salida para un nuevo punto de datos, sigue el árbol de decisiones basado en las características de ese punto de datos y termina en una de las hojas. El valor predicho es simplemente la clase mayoritaria de los puntos de datos en esa hoja.

Arboles de clasificación

Ejemplo de construcción del árbol:

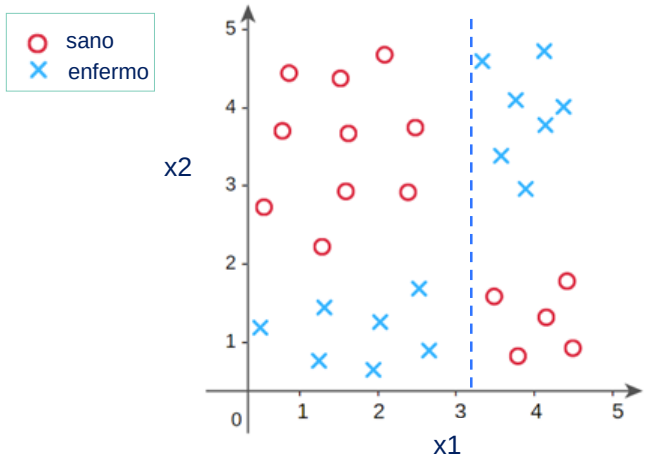


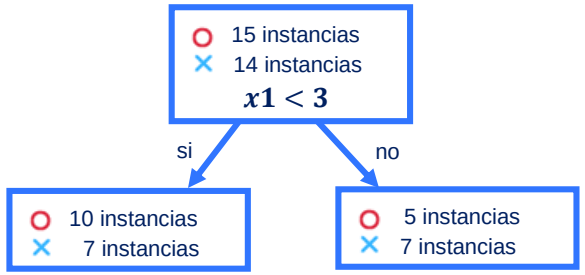
Nodo Raíz

15 instancias
14 instancias

Arboles de clasificación

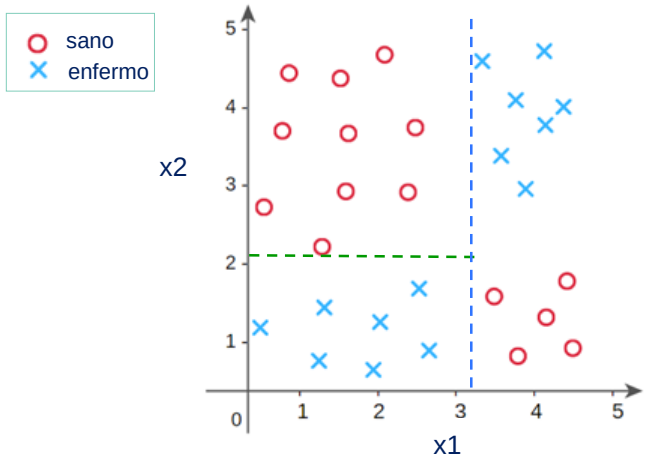
Ejemplo de construcción del árbol:

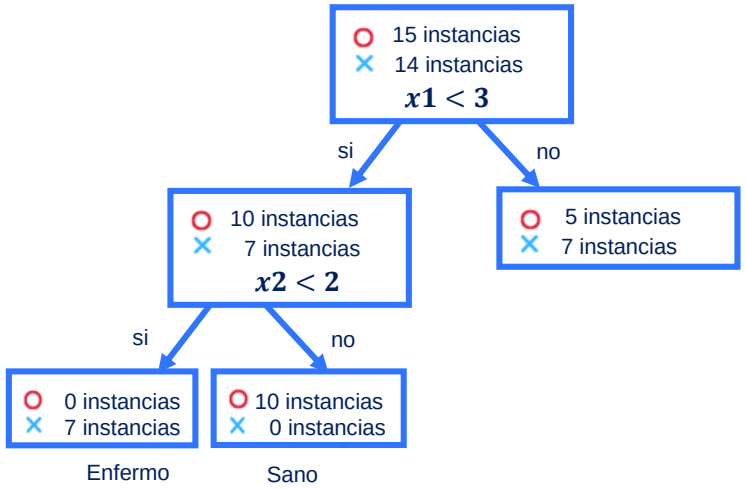




Arboles de clasificación

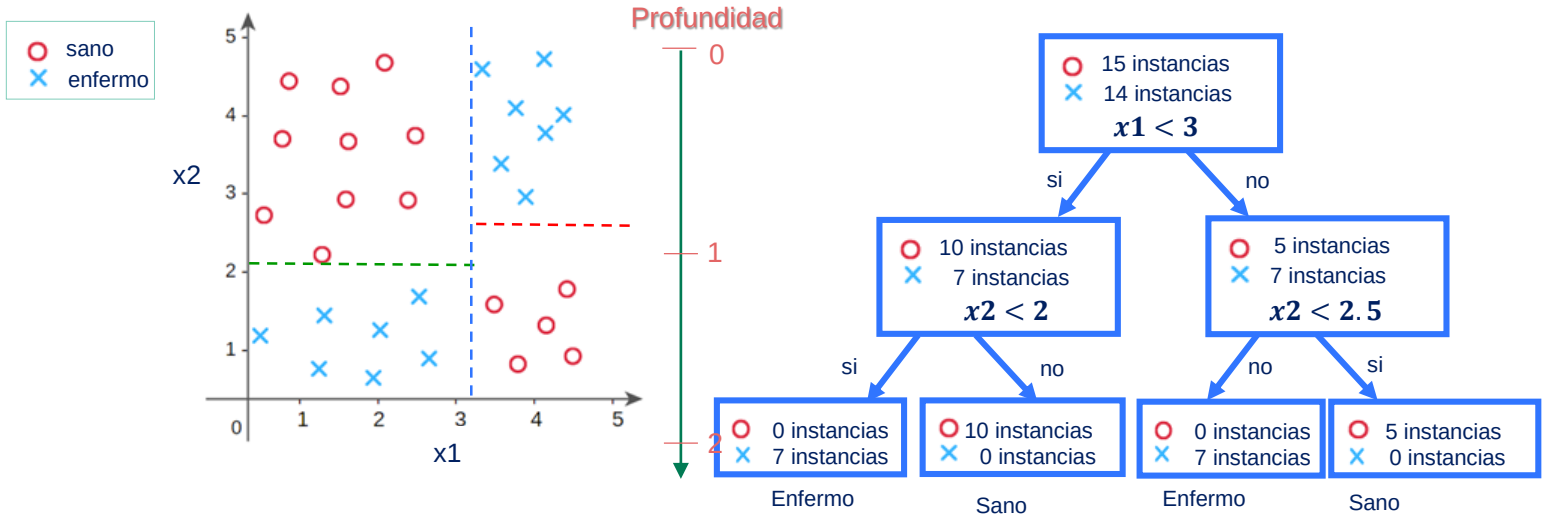
Ejemplo de construcción del árbol:





Arboles de clasificación

Ejemplo de construcción del árbol:



Arboles de clasificación

Medidas de impureza

$$\boldsymbol{Gini} = 1 - \sum_{i=1}^{c} (p_i)^2$$

$$= 1 - (probabilidad\ clase\ 1)^2 - (probabilidad\ clase\ 2)^2$$

p(🔴) = 1 p(🟢) =0

$$1 - \left(\frac{10}{10}\right)^2 - \left(\frac{0}{10}\right)^2 = 0$$

$$1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2 = 0.4$$

$$1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2 = 0.5$$

$$1 - \left(\frac{3}{10}\right)^2 - \left(\frac{3}{10}\right)^2 - \left(\frac{4}{10}\right)^2 = 0.7$$

$$1 - \left(\frac{1}{10}\right)^2 - \cdots - \left(\frac{1}{10}\right)^2 \approx 1$$

$Gini = 0$ Igualdad total

$Gini = 1$ Max. desigualdad

$$\boldsymbol{Entropy} = \sum_{i=1}^{c} -p_i \log p_i$$

$Entropia = 0$

$Entropia = 0.8$

$Entropia = 1$

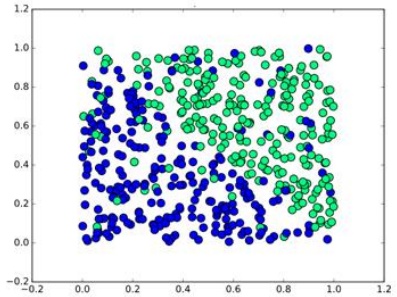
$Entropy = 0$ Orden total

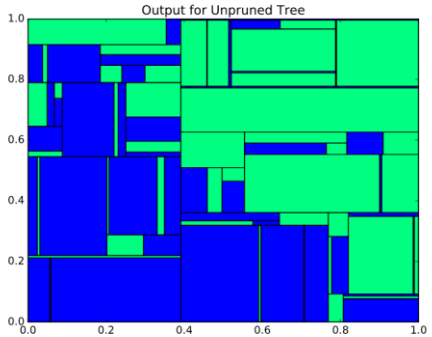
$Entropy = 1$ Max. desorden

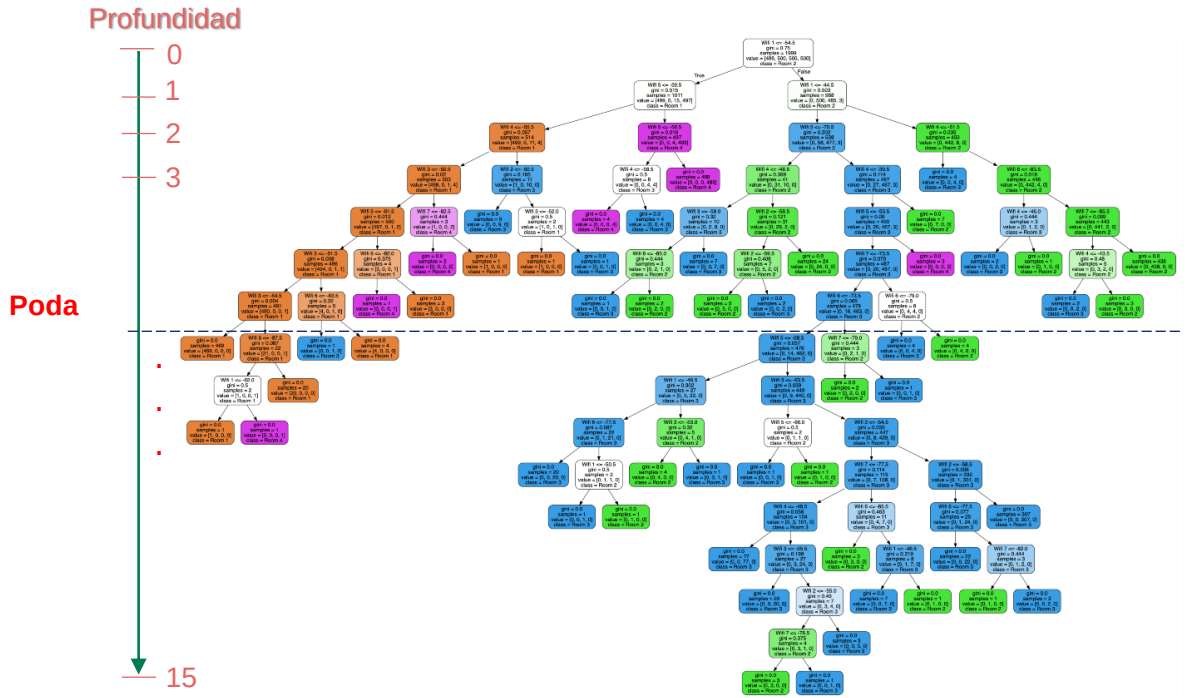
Arboles de clasificación

- Parámetros importantes

❖ max_Depth = {None} → hasta que las hojas sean puras o tengan un mínimo de muestras



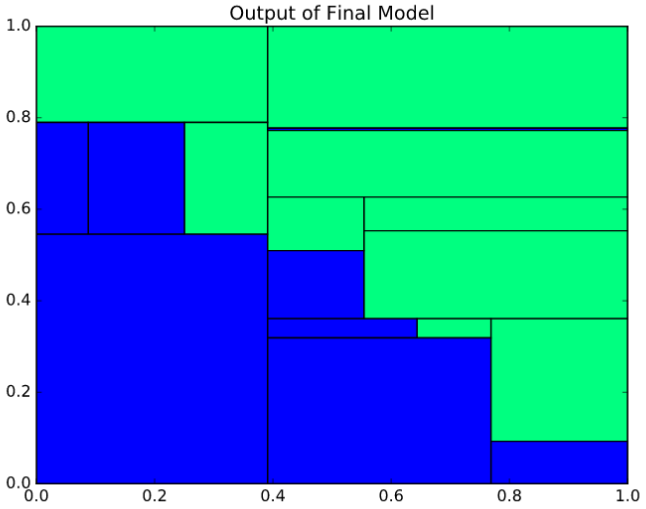


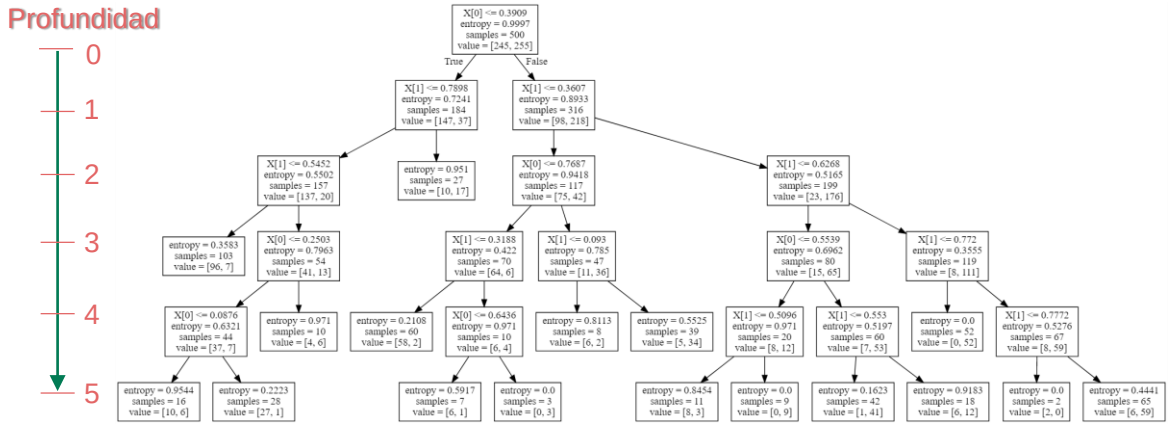


Arboles de clasificación

- Regularización por poda

max_Depth = **5** ⟶ Se genera el árbol hasta una profundidad máxima de **5**





Bibliografía

- J. Watt and R. Borhani and A. Katsaggelos (2020). Machine Learning Refined: Foundations, Algorithms, and Applications. 2nd Edition. Cambridge: Cambridge University Press.
- C. Bishop (2006). Pattern Recognition and Machine Learning. Springer, New York.
- S. Raschka & V. Mirjalili (2019). *Python Machine Learning*. Third Edition. California: O'Reilly Media.

Sugerencia de links interesantes:

DERIVADAS - Clase Completa: Explicación Desde Cero
https://www.youtube.com/watch?v=_6-zwdrqD3U

DERIVADAS: Las Famosas Reglas EXPLICADAS
https://www.youtube.com/watch?v=O6PeN5SJxzk

¡Gracias!