



Programa de formación
**MACHINE LEARNING
AND DATA SCIENCE MLDS**

Facultad de
INGENIERÍA





Módulo 2

Introducción al Machine

Learning con *Python*

Unidad 3

Aprendizaje Supervisado: Métodos de Clasificación

Clase sincrónica

Facultad de
INGENIERÍA



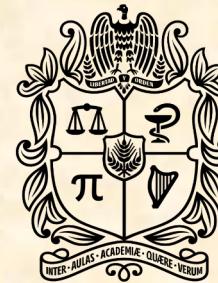


Bienvenida

Fabio Augusto Gonzalez, PhD.

<https://dis.unal.edu.co/~fgonza/>

fagonzalezo@unal.edu.co



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Departamento de Ingeniería de Sistemas e Industrial
Facultad de Ingeniería
Universidad Nacional de Colombia
Sede Bogotá



Tabla de contenidos

- 1 Métodos de Clasificación
- 2 Clasificador Naïve Bayes
- 3 Clasificador *K Nearest Neighbors* (KNN)
- 4 Árboles de Decisión
- 5 Bosques Aleatorios
- 6 Máquina de Vectores de Soporte (SVM)
- 7 Red Neuronal Artificial
- 8 Series de tiempo

Objetivos de aprendizaje



Unidad 3 - Aprendizaje supervisado: métodos de clasificación

Al finalizar la unidad usted deberá ser capaz de:



1

Conocer los fundamentos de diferentes métodos de clasificación.



2

Implementar con ayuda de la librería *scikit-learn* diferentes modelos de clasificación.



3

Conocer los fundamentos de los modelos de regresión no lineal mediante redes neuronales.

Objetivos de aprendizaje



Unidad 3 - Aprendizaje supervisado: métodos de clasificación

Al finalizar la unidad usted deberá ser capaz de:



4

Conocer los fundamentos de los modelos de análisis de series de tiempo mediante redes neuronales.

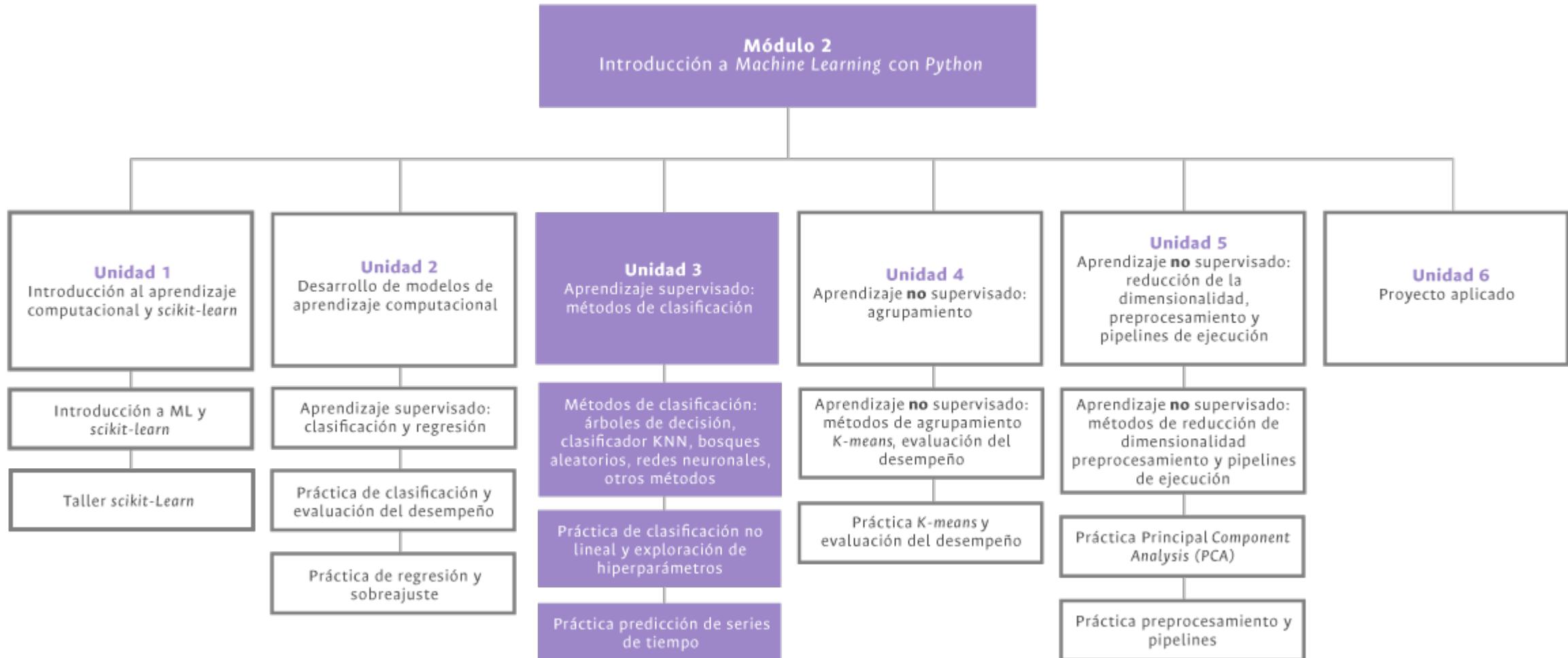


5

Afinar modelos de *Machine Learning* mediante la exploración de hiperparámetros.

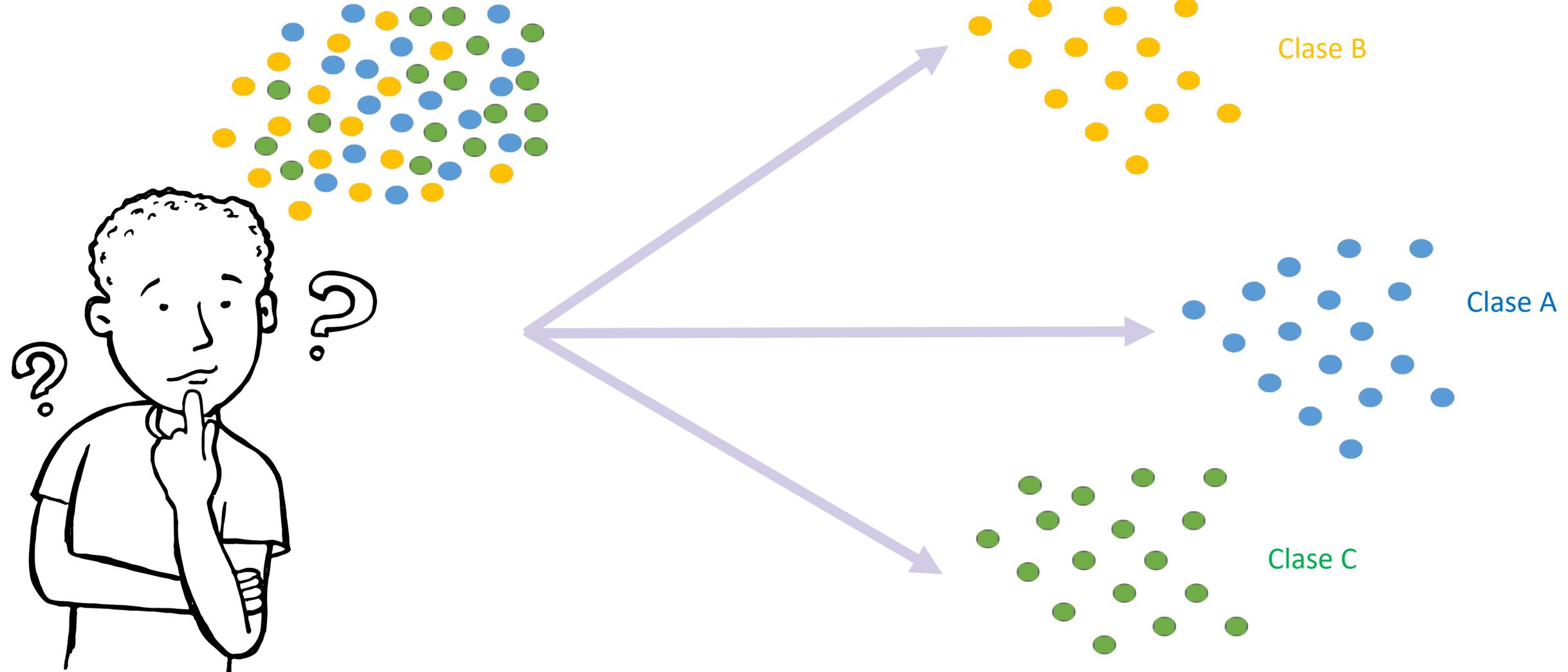


Mapa de contenidos de la unidad



1

Métodos de Clasificación



Métodos de Clasificación

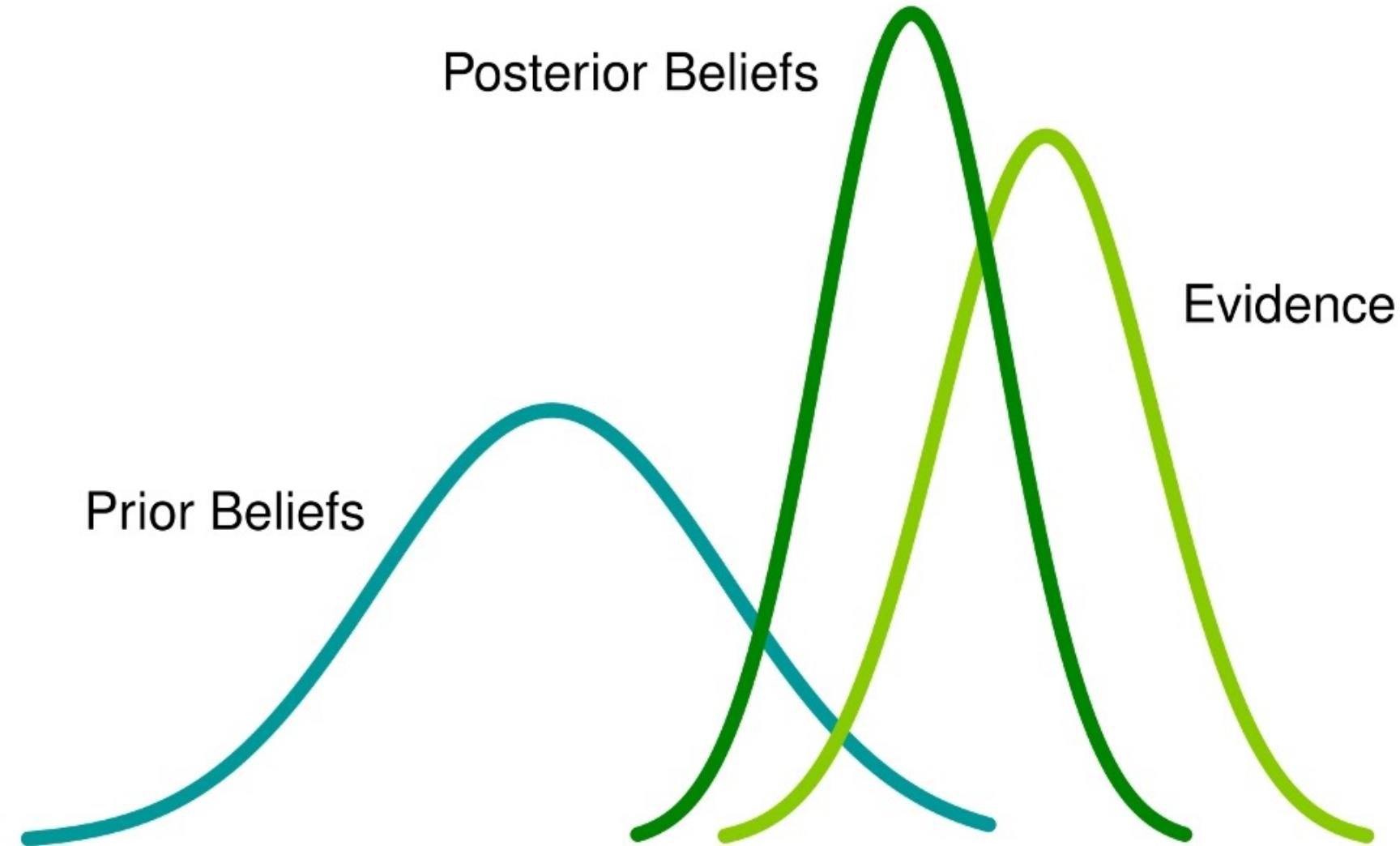


El aprendizaje computacional comprende una familia de métodos muy grande, los cuales poseen múltiples y variados enfoques.

Enfoque	Método
Probabilístico	Clasificador ingenuo de Bayes
Bioinspirado	Redes neuronales
Geométrico	Máquina de vectores de soporte
Basado en memoria	Clasificador de los K vecinos más cercanos
Basado en árboles	Árboles de decisión Bosques aleatorios

2

Clasificador Naïve Bayes



Clasificador Naïve Bayes

Clasificadores Bayesianos

Clasificadores Bayesianos

Corresponden a un conjunto de clasificadores con un enfoque probabilístico al aprendizaje computacional. Estos hacen un uso extensivo del teorema de Bayes.

El clasificador ingenuo de Bayes (Naïve Bayes) es uno de los más sencillos de esta familia.

Clasificador Naïve Bayes

 Teorema de Bayes

Teorema de Bayes

El teorema de Bayes provee una manera de calcular la probabilidad posterior a partir de la verosimilitud y las probabilidades previas.

$$P(C_i|\mathbf{x}) = \frac{P(\mathbf{x}|C_i)P(C_i)}{P(\mathbf{x})}$$

Probabilidad *posterior* ← P($C_i|\mathbf{x}$) ↑ Verosimilitud (Likelihood) → Probabilidad *previa* de clases
↓ ↓ Probabilidad *previa* de características

Clasificador Naïve Bayes

 Entrenamiento

El **entrenamiento** de un modelo ingenuo de Bayes consiste en estimar los parámetros necesarios para calcular:

- Probabilidad previa $P(C_i)$ para cada clase (prior)
- Verosimilitud $P(x|C_i)$ para cada par de característica y clase (likelihood)

Clasificador Naïve Bayes

 Predicción

El clasificador ingenuo estima la probabilidad posterior de un ejemplo con características x_1, x_2, \dots, x_m de la siguiente manera:

$$P(C_i|\mathbf{x}) \sim P(x_1|C_i)P(x_2|C_i)\dots P(x_m|C_i)P(C_i)$$

Luego, la clase con la mayor probabilidad posterior es seleccionada como la predicción. El denominador de fórmula del teorema de Bayes es ignorado, pues no afecta la selección.



El clasificador ingenuo de Bayes recibe su nombre porque asume la independencia de las características de un ejemplo dada una clase C_i , suposición la cual puede que no se cumpla. Sin embargo, el clasificador ingenuo puede obtener resultados sorprendentemente buenos

Clasificador Naïve Bayes

 Ejemplo

- Se busca predecir el atributo de clase *Manejar Bicicleta* utilizando las características Clima, Temperatura, Humedad y Viento.
- Se utiliza el conjunto de datos de entrenamiento para calcular tablas de frecuencia y verosimilitud.
- También se calcula la probabilidad previa de cada clase, *Manejar* y *No Manejar*.

Clima	Temperatura	Humedad	Ventoso	Manejar bicicleta
Lluvioso	Caliente	Alta	No	No
Lluvioso	Caliente	Alta	Sí	No
Nublado	Caliente	Alta	No	Sí
Soleado	Tibio	Alta	No	Sí
Soleado	Frío	Normal	No	Sí
Soleado	Frío	Normal	Sí	No
Nublado	Frío	Normal	Sí	Sí
Lluvioso	Tibio	Alta	No	No
Lluvioso	Frío	Normal	No	Sí
Soleado	Tibio	Normal	No	Sí
Lluvioso	Tibio	Normal	Sí	Sí
Nublado	Tibio	Alta	Sí	Sí
Nublado	Caliente	Normal	No	Sí
Soleado	Tibio	Alta	Sí	No

Clasificador Naïve Bayes

 Ejemplo: fase de entrenamiento

- Cálculo de probabilidad previa:

$$P(\text{Manejar}) = \frac{\text{Ejemplos manejar}}{\text{Total ejemplos}} = \frac{9}{14}$$

$$P(\text{No manejar}) = \frac{\text{Ejemplos no manejar}}{\text{Total ejemplos}} = \frac{5}{14}$$

- Cálculo de la verosimilitud

$$P(\text{Clima} = \text{soleado} | \text{Manejar}) = \frac{\text{Ejemplos soleado y manejar}}{\text{Ejemplos manejar}} = \frac{3}{9}$$

- Primero se calcula la tabla de frecuencia y con ella la tabla de verosimilitud para cada característica

		Manejar	
		Sí	No
Clima	Soleado	3	2
	Nublado	4	0
	Lluvioso	2	3

→

		Manejar	
		Sí	No
Clima	Soleado	3/9	2/5
	Nublado	4/9	0/5
	Lluvioso	2/9	3/5

Clasificador Naïve Bayes

 Ejemplo: fase de entrenamiento

Frecuencia por cada característica

		Manejar	
		Sí	No
Clima	Soleado	3	2
	Nublado	4	0
	Lluvioso	2	3

Verosimilitud por cada característica

		Manejar	
		Sí	No
Clima	Soleado	3/9	2/5
	Nublado	4/9	0/5
	Lluvioso	2/9	3/5

Frecuencia por cada característica

		Manejar	
		Sí	No
Temp.	Caliente	2	2
	Tibio	4	2
	Frío	3	1

Verosimilitud por cada característica

		Manejar	
		Sí	No
Temp.	Caliente	2/9	2/5
	Tibio	4/9	2/5
	Frío	3/9	1/5

		Manejar	
		Sí	No
Humedad	Alta	3	4
	Normal	6	1

		Manejar	
		Sí	No
Humedad	Alta	3/9	4/5
	Normal	6/9	1/5

		Manejar	
		Sí	No
Ventoso	No	6	2
	Sí	3	3

		Manejar	
		Sí	No
Ventoso	No	6/9	2/5
	Sí	3/9	3/5

Clasificador Naïve Bayes

 Ejemplo: fase de predicción

- Se clasifica el siguiente ejemplo:

Clima	Temperatura	Humedad	Ventoso	Manejar
Lluvioso	Frío	Alta	Sí	?

- Se calcula la probabilidad posterior para “Sí” y para “No”

$$P(\text{Sí}|\mathbf{x}) = P(\text{Lluvioso|Sí}) \cdot P(\text{Frío|Sí}) \cdot P(\text{Alta|Sí}) \cdot P(\text{Ventoso|Sí}) \cdot P(\text{Sí})$$

$$P(\text{Sí}|\mathbf{x}) = \frac{2}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{3}{9} \cdot \frac{9}{14} = 0.00529$$

$$P(\text{No}|\mathbf{x}) = P(\text{Lluvioso|No}) \cdot P(\text{Frío|No}) \cdot P(\text{Alta|No}) \cdot P(\text{Ventoso|No}) \cdot P(\text{No})$$

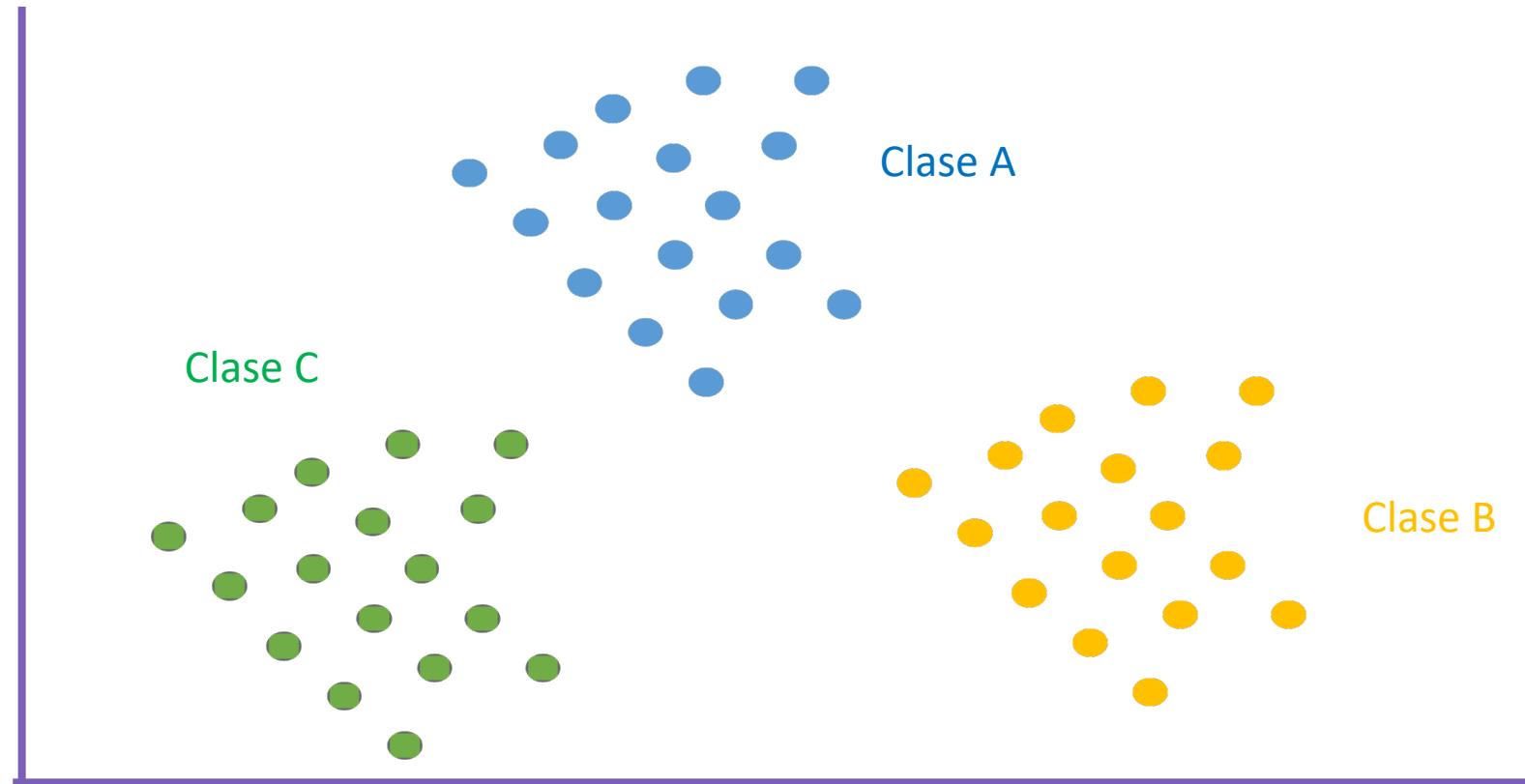
$$P(\text{No}|\mathbf{x}) = \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{4}{5} \cdot \frac{3}{5} \cdot \frac{5}{14} = 0.02507$$

- Se selecciona la categoría con mayor probabilidad posterior

La clase con mayor probabilidad posterior es No.

3

Clasificador *K Nearest Neighbors* (KNN)



Clasificador KNN

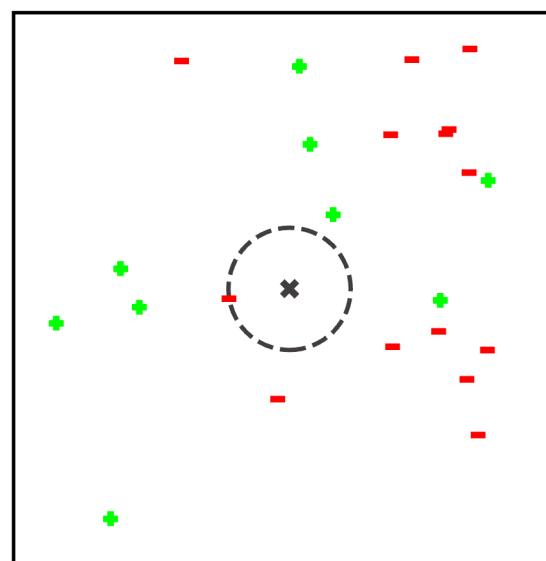
 Aprendizaje basado en instancias

- Pertenecen a una familia de algoritmos de aprendizaje denominada aprendizaje basado en instancias o aprendizaje basado en memoria
- Estos algoritmos no intentan construir un modelo generalizador de manera explícita, sino que simplemente memorizan los ejemplos de entrenamiento.

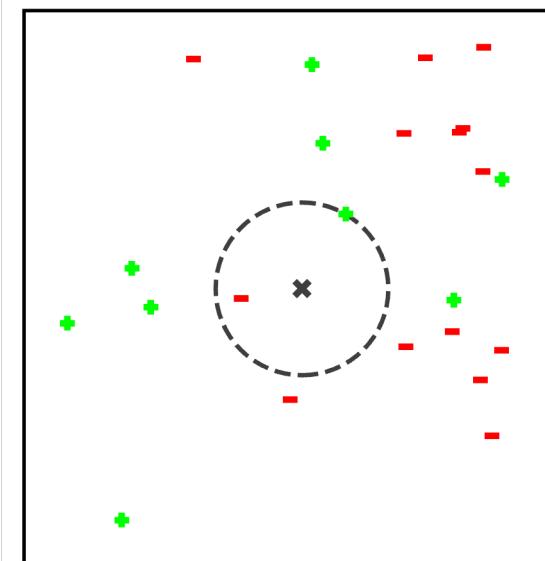
Clasificador KNN

Vecinos más cercano

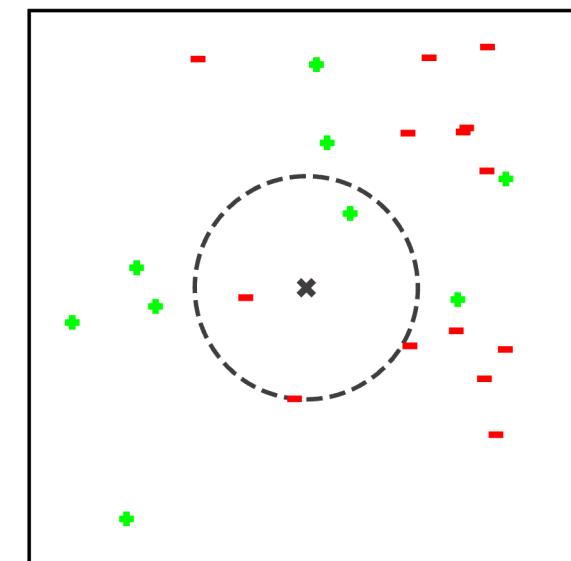
Los k ésimos vecinos más cercanos de un ejemplo x son los puntos de datos con la k éSIMA distancia más pequeña a x .



Primer vecino
más cercano



Segundo vecino
más cercano

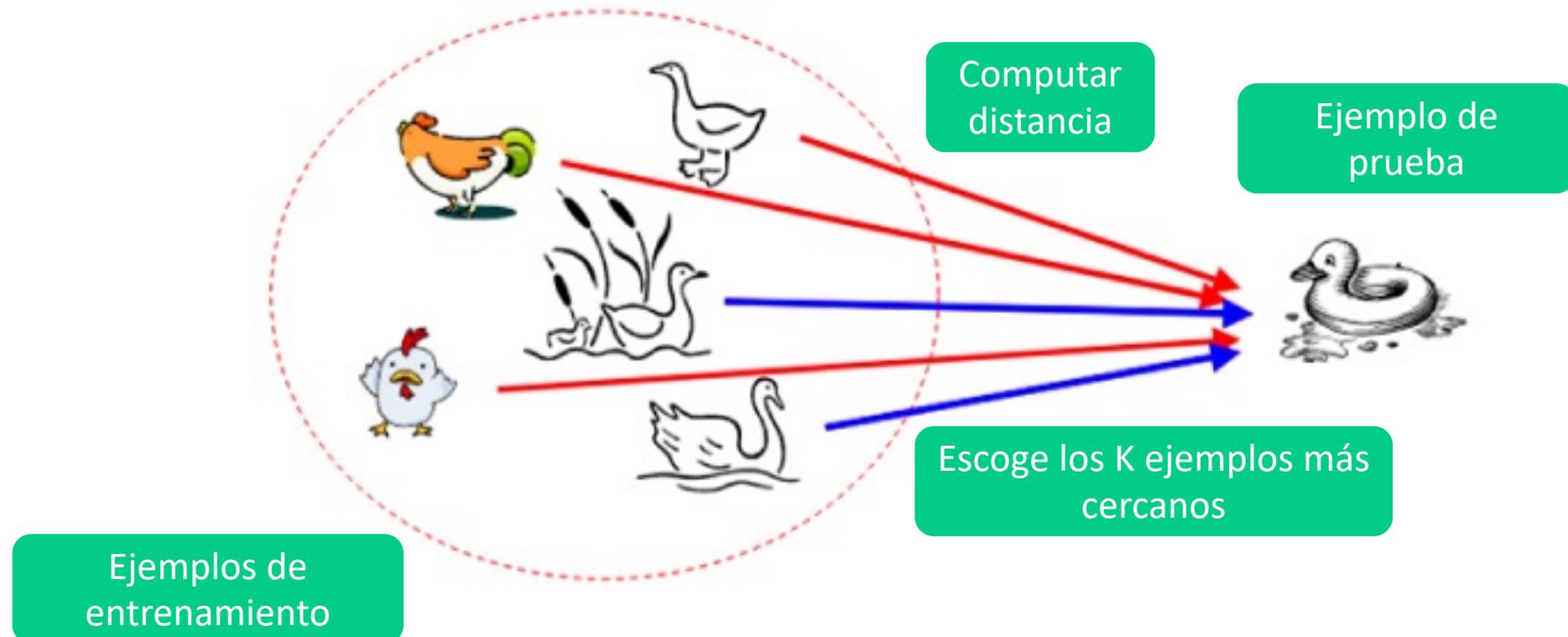


Tercer vecino
más cercano

Clasificador KNN

Idea básica

Si camina como un pato y suena como un pato, entonces, probablemente es un pato

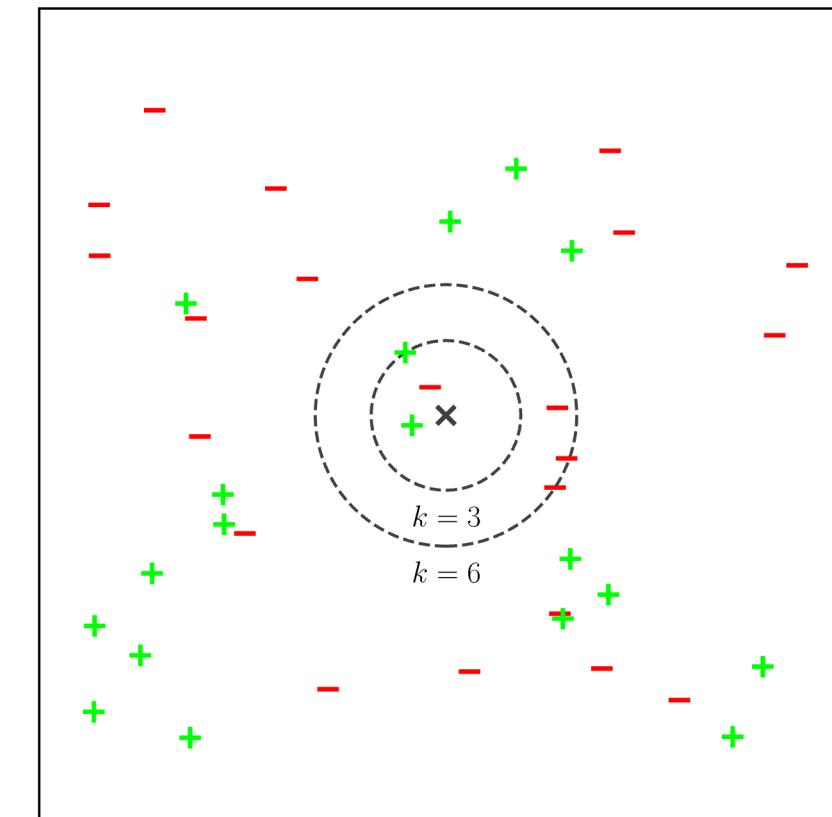


Clasificador KNN

Proceso de predicción

Para realizar el proceso de predicción se necesitan lo siguientes:

- El conjunto de ejemplos de entrenamiento.
- Una métrica de distancia que compute una distancia entre ejemplos.
- El valor k , el número de vecinos más cercanos a identificar.

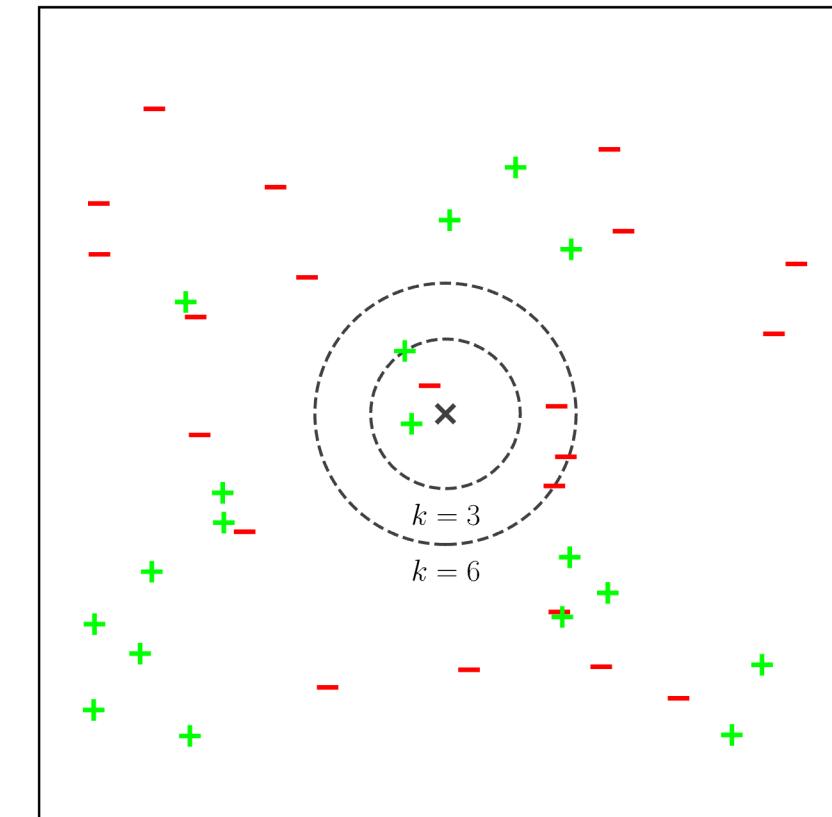


Clasificador KNN

Proceso de predicción

Para clasificar un ejemplo nuevo:

- Se calcula la distancia del nuevo ejemplo a todos los ejemplos de entrenamiento.
- Se identifican los k vecinos más cercanos.
- Se utilizan las etiquetas de los vecinos más cercanos para determinar la etiqueta o clase del nuevo ejemplo, a través de un voto de mayorías.

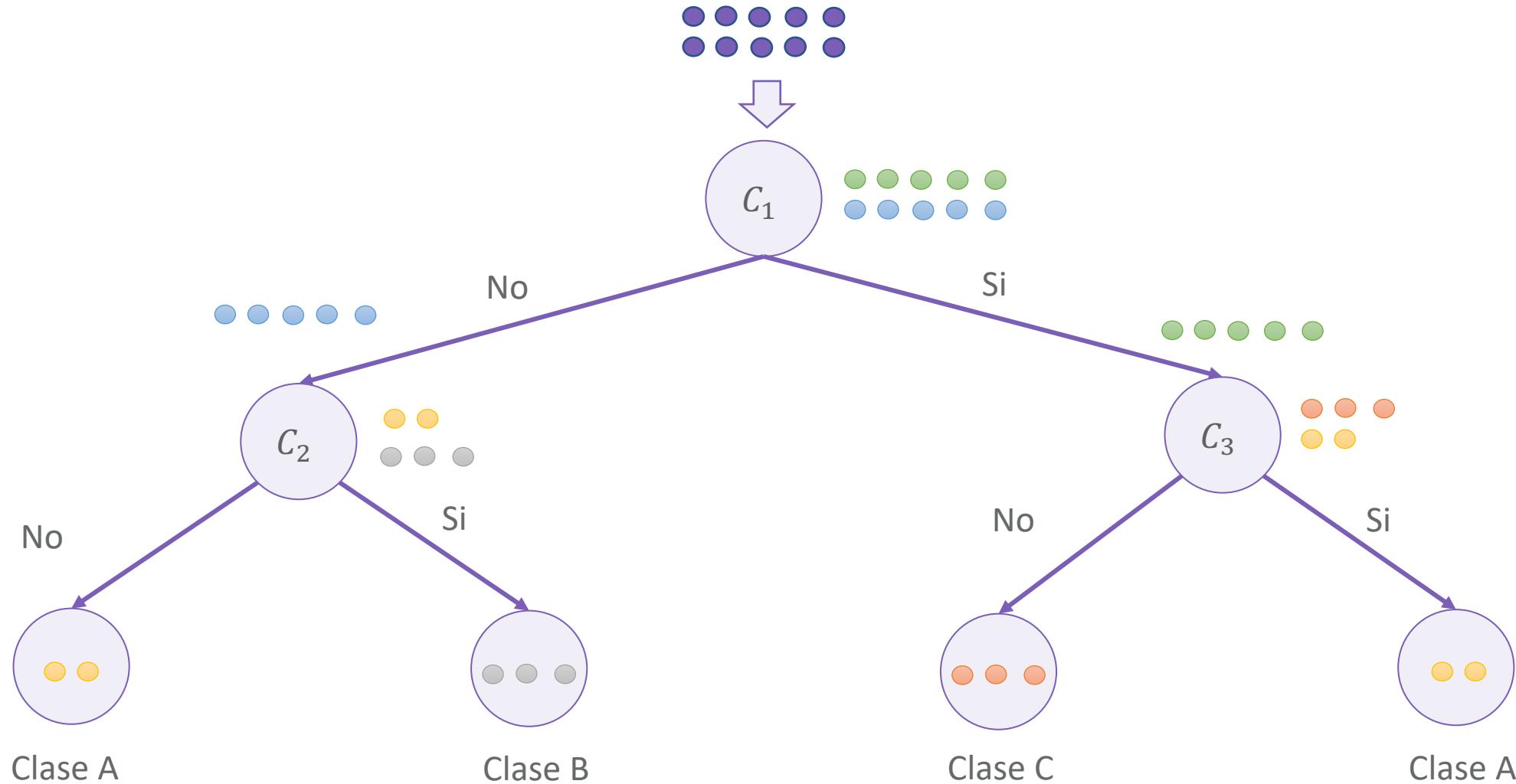


Ejemplo de clasificador KNN

En este ejemplo para $k=3$ se predice la clase positiva y para $k=6$ se predice la clase negativa

4

Árboles de Decisión

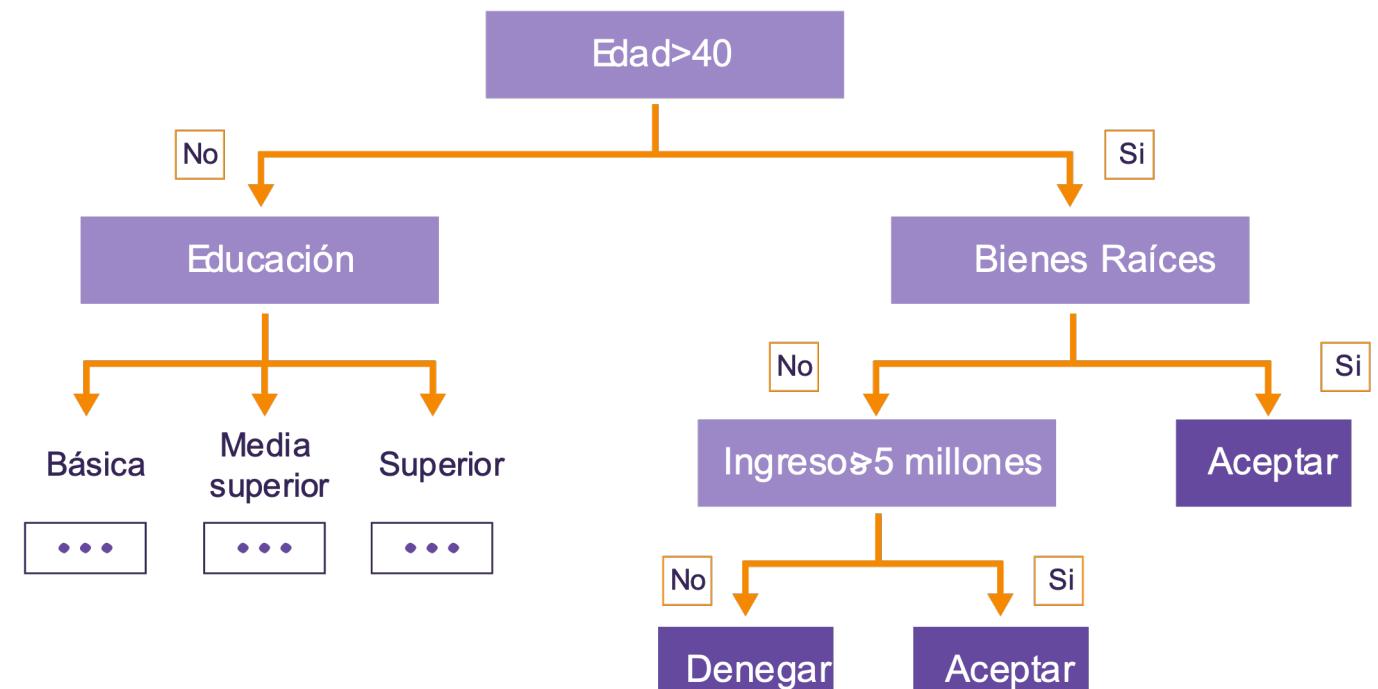


Árboles de Decisión

Estructura

- Los árboles de decisión son modelos de clasificación y regresión estructurados como árboles.
- Un árbol está compuesto por nodos de **decisión** y nodos **terminales u hojas**.
- El nodo de decisión en la posición más alta es llamado **nodo raíz**.

Árbol de decisión para la otorgación de préstamos



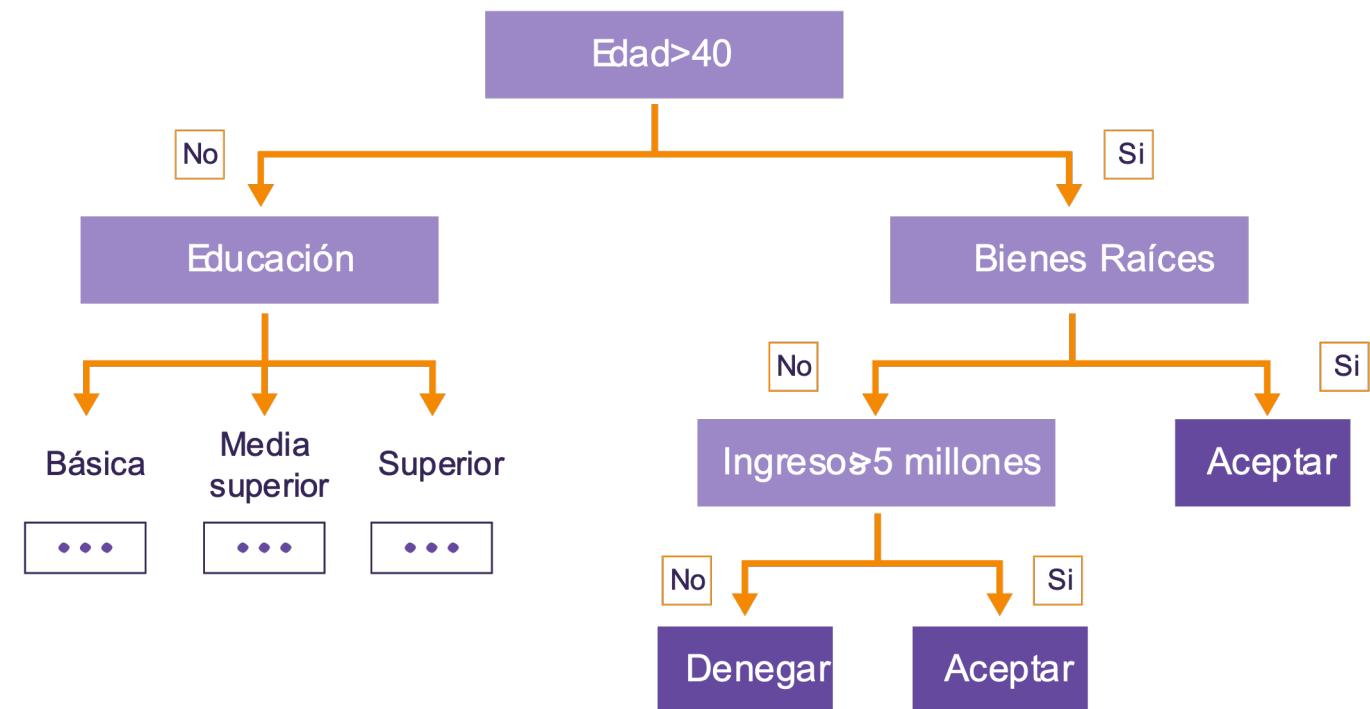
Árboles de Decisión

Estructura

Cada **nodo de decisión** tiene las siguientes propiedades:

- Se asocia con una característica y una regla de separación de datos.
- Tiene dos o más ramificaciones en otros nodos de decisión u hojas. Estas ramificaciones corresponden a valores específicos de la característica asociada con el nodo.
- Los nodos terminales representan una clasificación.

Árbol de decisión para la otorgación de préstamos



Árboles de Decisión

Proceso de Clasificación

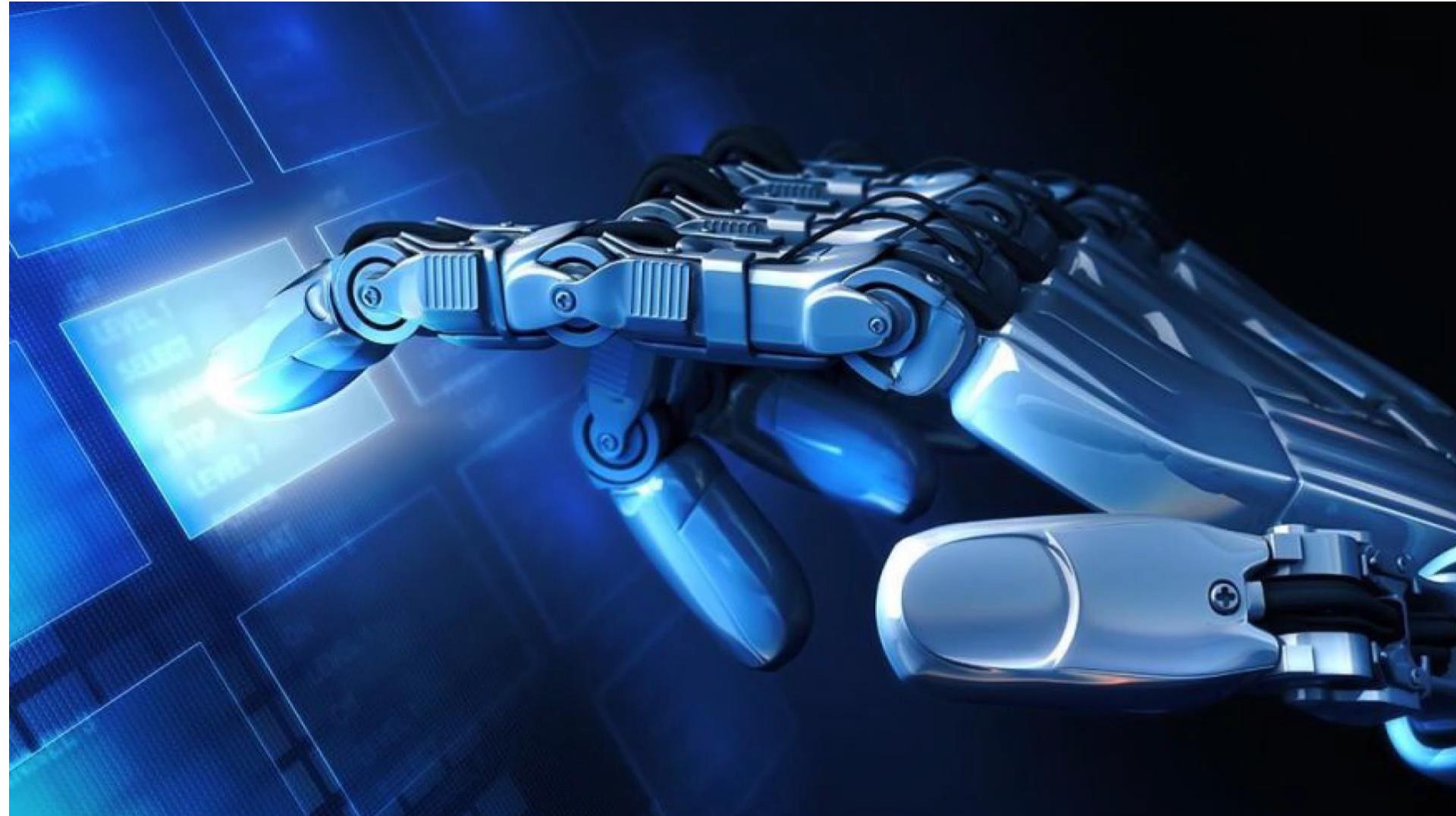
El proceso de clasificación se realiza haciendo un recorrido desde la raíz hasta una hoja.

En este recorrido se sigue el camino dictado por la regla de decisión de cada nodo de decisión alcanzado.

Los árboles de decisión son modelos altamente populares debido a su interpretabilidad..

Árboles de Decisión

Proceso de Clasificación (Video)



Árboles de Decisión

Algoritmo de entrenamiento de árboles de decisión

Algoritmo:

1. Encontrar el atributo con mayor ganancia de información y crear un nodo de decisión con este.
2. Separar los datos de entrenamiento, de acuerdo con el valor que tengan del atributo correspondiente.
3. Para cada partición:
 - a. Si todos los datos son de la misma clase poner un nodo hoja.
 - b. Si no aplicar el mismo procedimiento de manera recursiva en la partición.

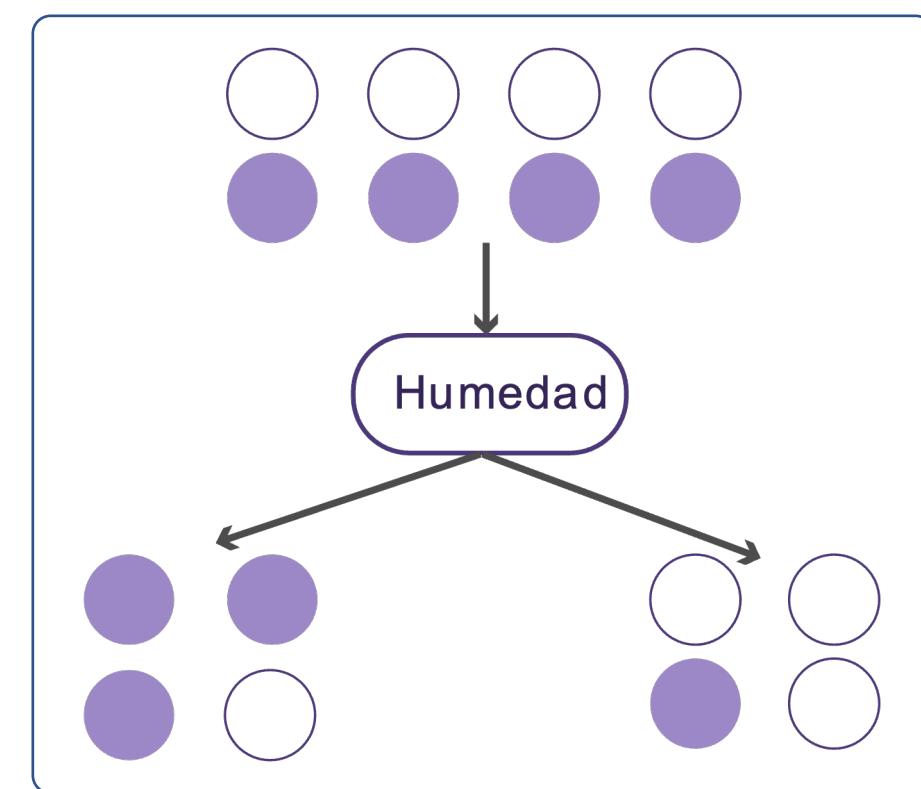
El componente más importante de este algoritmo es la estimación de la ganancia de información de un atributo. Esta ganancia de información se relaciona con el ‘orden’ que un atributo induce en un conjunto de datos.

Árboles de Decisión

Ganancia de Información

La ganancia de información de un atributo está relacionada con la homogeneidad de los grupos en los que se separa un conjunto de datos.

En la figura se muestra un conjunto inicial de datos, en los cuales cada mitad tiene una clase diferente. Esta es la heterogeneidad máxima que puede tener un conjunto. Cuando usamos el atributo *Humedad* para separarlos, se crean dos grupos, en cada uno de estos la relación de elementos de una clase a la otra es de $\frac{3}{4}$ y $\frac{1}{4}$, es decir, los grupos resultantes son menos heterogéneos y, por lo tanto, conocer este atributo ofrece una ganancia de información.



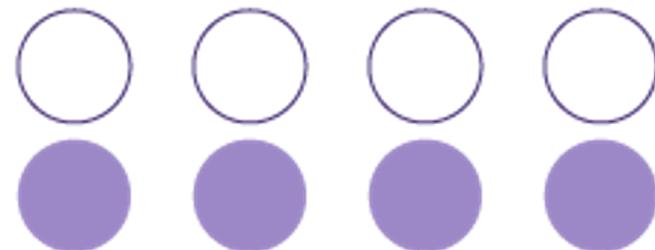
Árboles de Decisión

Entropía de Shannon

Una forma de medir la heterogeneidad de un conjunto es usando la entropía definida de la siguiente manera:

$$S = - \sum_{i=1}^N p_i \log_2 p_i,$$

Donde p_i es la frecuencia de un ejemplo de ser de la clase i.



En este conjunto de datos la entropía es 1:

$$S = - \left(\frac{1}{2} \log_2 \left(\frac{1}{2} \right) + \frac{1}{2} \log_2 \left(\frac{1}{2} \right) \right) = 1$$

Árboles de Decisión

Entropía de Shannon

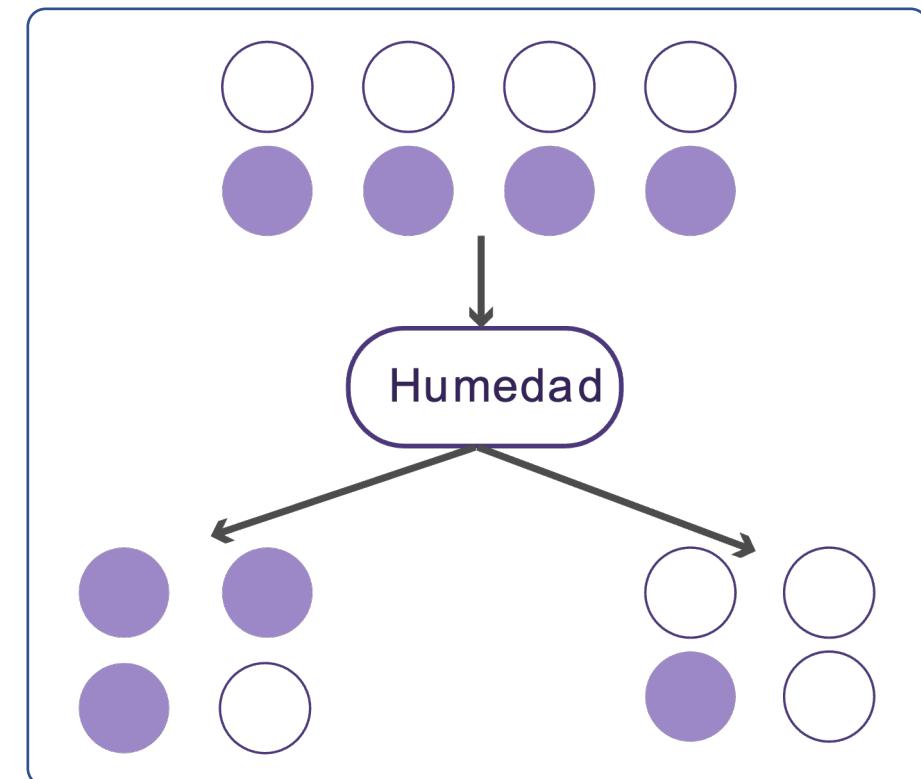
La ganancia de información es la diferencia de la entropía antes y después de una partición: se define de la siguiente manera:

$$IG(Q) = S_O - \sum_{i=1}^q \frac{N_i}{N} S_i,$$

La entropía después de la partición se calcula como un promedio ponderado por el tamaño de los grupos resultantes.

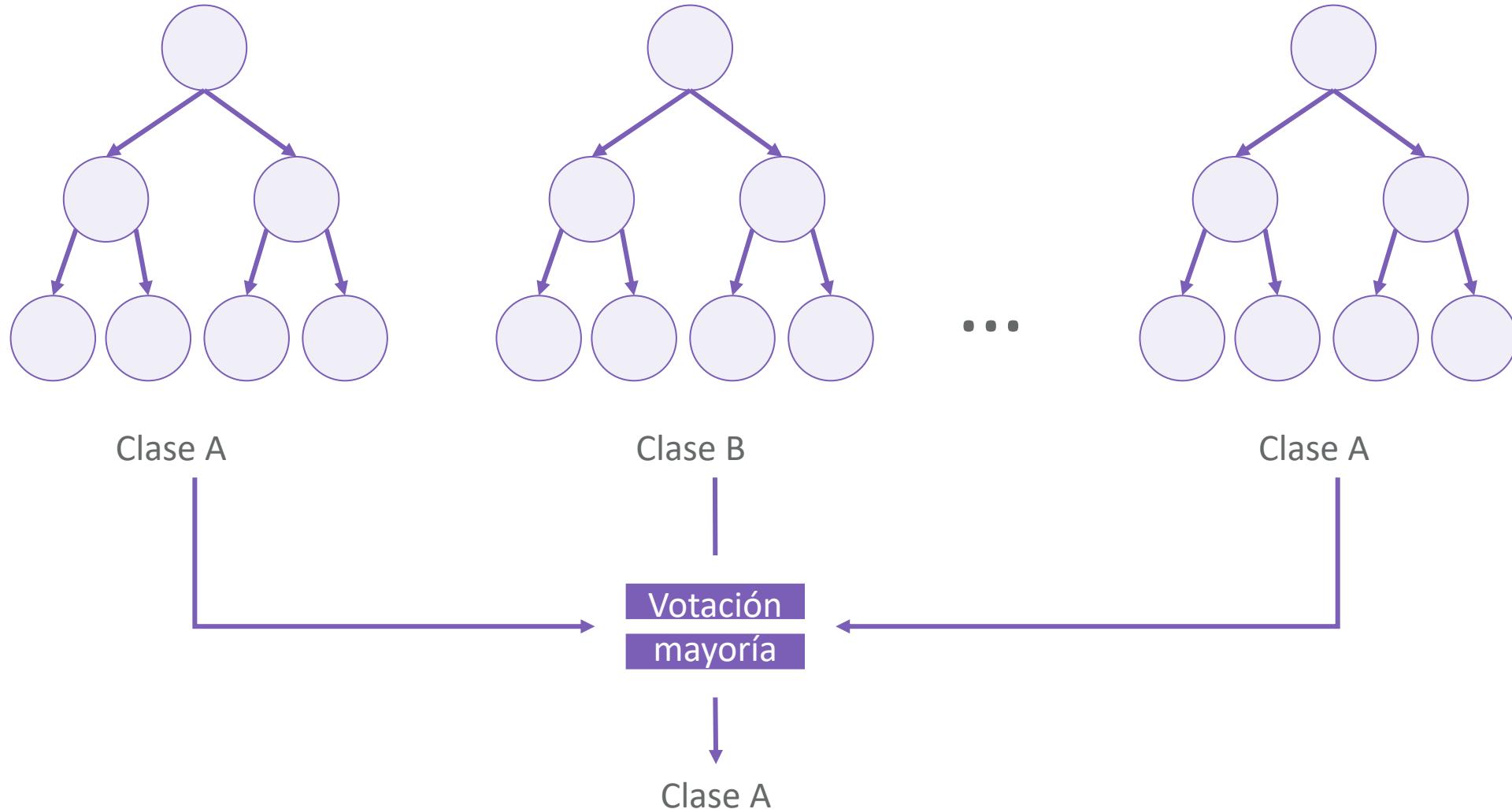
En este caso cada grupo tiene el mismo tamaño y la entropía de cada grupo es la misma:

$$\begin{aligned} IG(Humedad) &= 1 - \left(\frac{3}{4} \log_2 \left(\frac{3}{4} \right) + \frac{1}{4} \log_2 \left(\frac{1}{4} \right) \right) \\ &= 1 - 0.811 \\ &= 0.188 \end{aligned}$$



5

Bosques Aleatorios



Bosques Aleatorios

 Ensambles

- Los métodos de ensamble usan un conjunto de múltiples modelos de clasificación para obtener un mejor desempeño del que podría ser obtenido por cualquiera de los modelos de clasificación de algoritmos de manera individual.
- Algunos métodos populares de ensamble son *Bagging*, *Boosting* y *Stacking*.
- El algoritmo de bosques aleatorios es un ensamble basado en *Bagging* (empaqueamiento) y árboles de decisión.

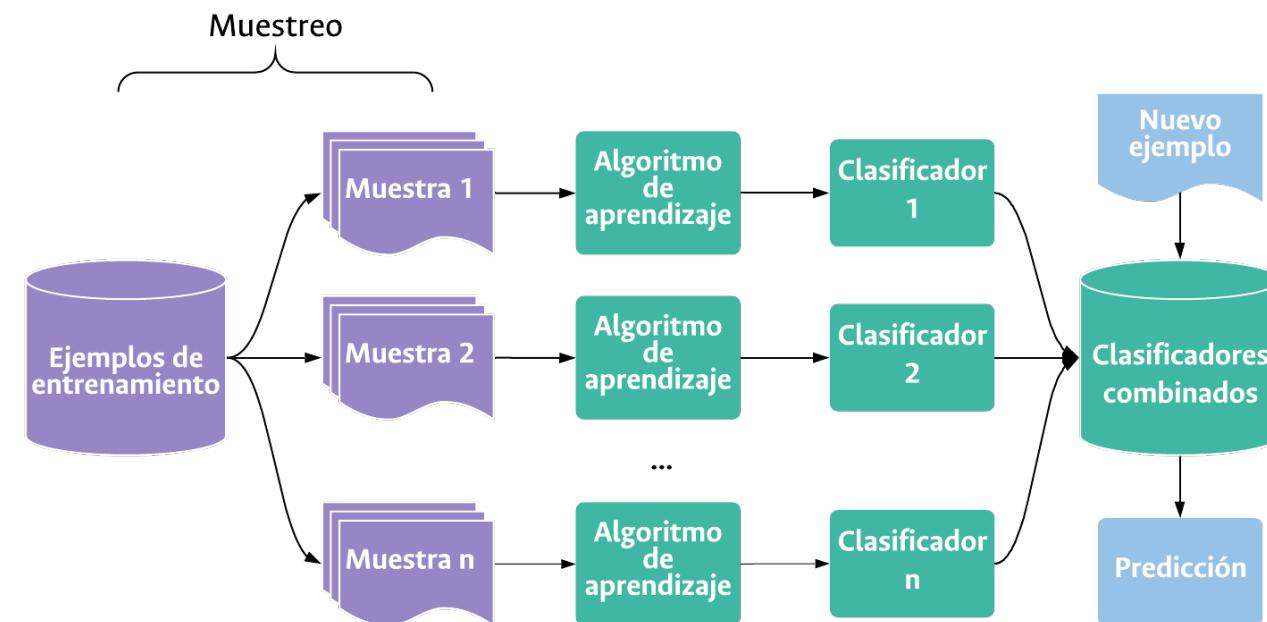
Bosques Aleatorios

Empaquetado (*Bagging*)

Al asumir un conjunto de datos X , el algoritmo realiza los siguientes pasos:

Muestreo (*Bootstrapping*)

Se forman n nuevos conjuntos de datos X_1, \dots, X_n , se muestrea *uniformemente* y con *reemplazo* de X



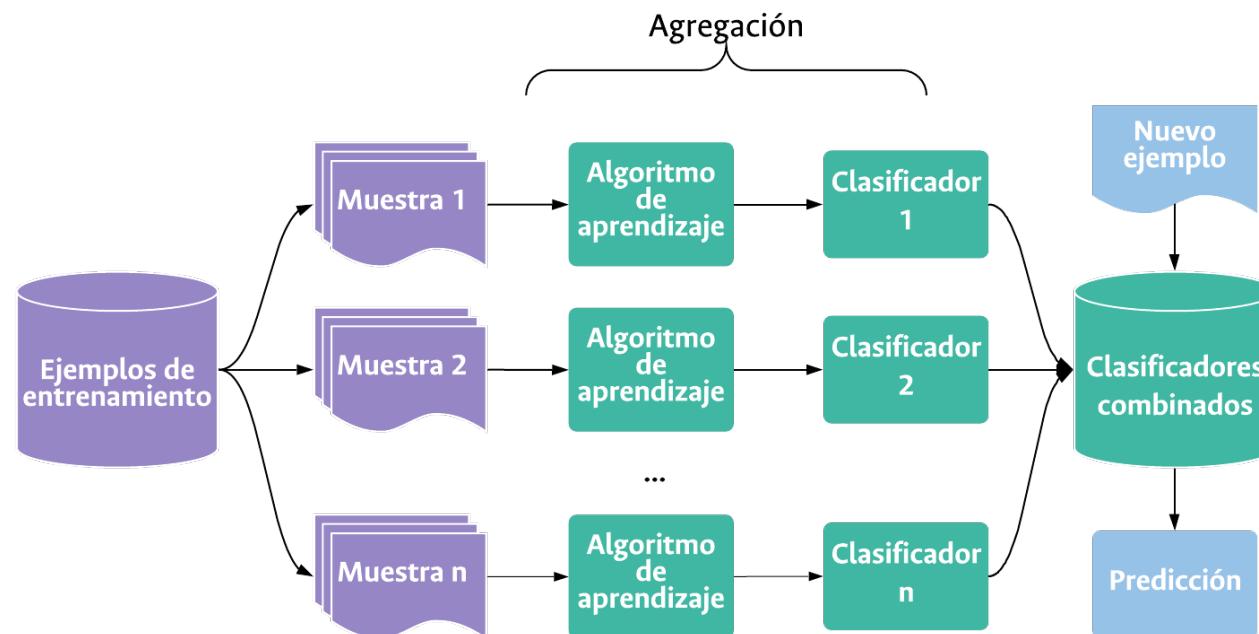
Esquema del algoritmo de ensambles empaquetado - muestreo

Bosques Aleatorios

Empaquetado (*Bagging*)

Agregación

Se entrenan n clasificadores distintos C_i sobre cada subconjunto X_i



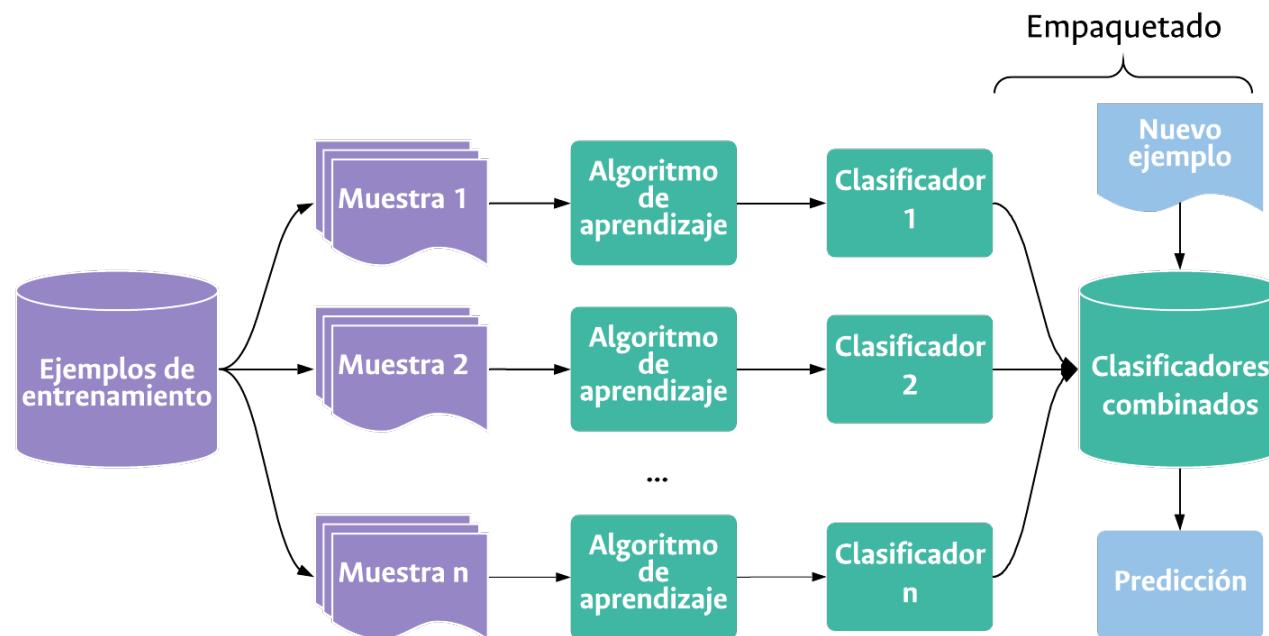
Esquema del algoritmo de ensambles empaquetado - agregación

Bosques Aleatorios

Empaquetado (*Bagging*)

Empaquetado

Los modelos son empaquetados y las nuevas muestras son clasificadas utilizando el voto de cada clasificador.



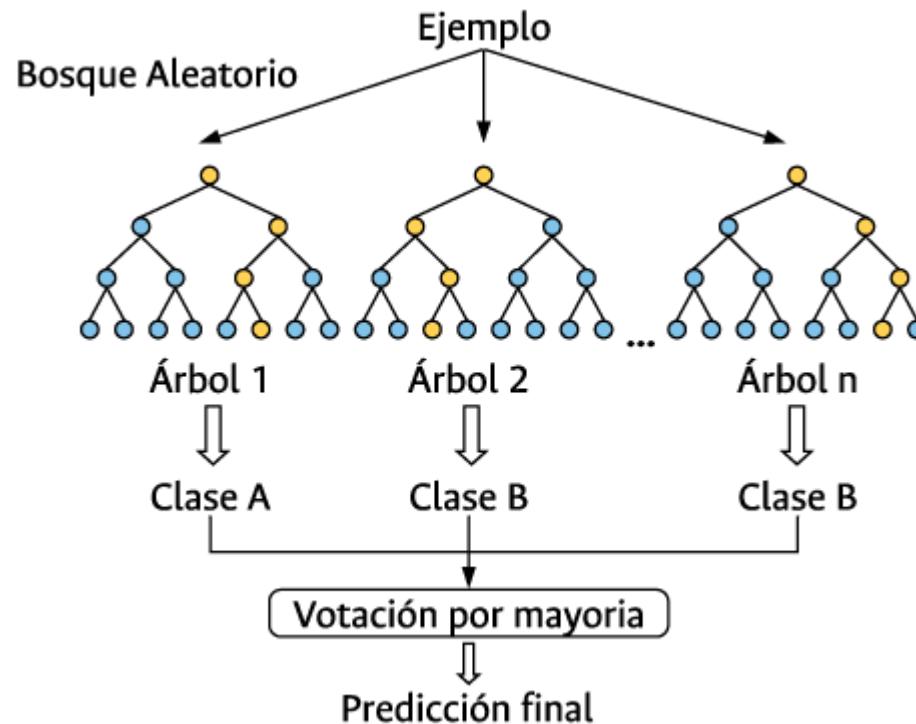
Esquema del algoritmo de ensambles empaquetado

Bosques Aleatorios

Empaquetado (*Bagging*)

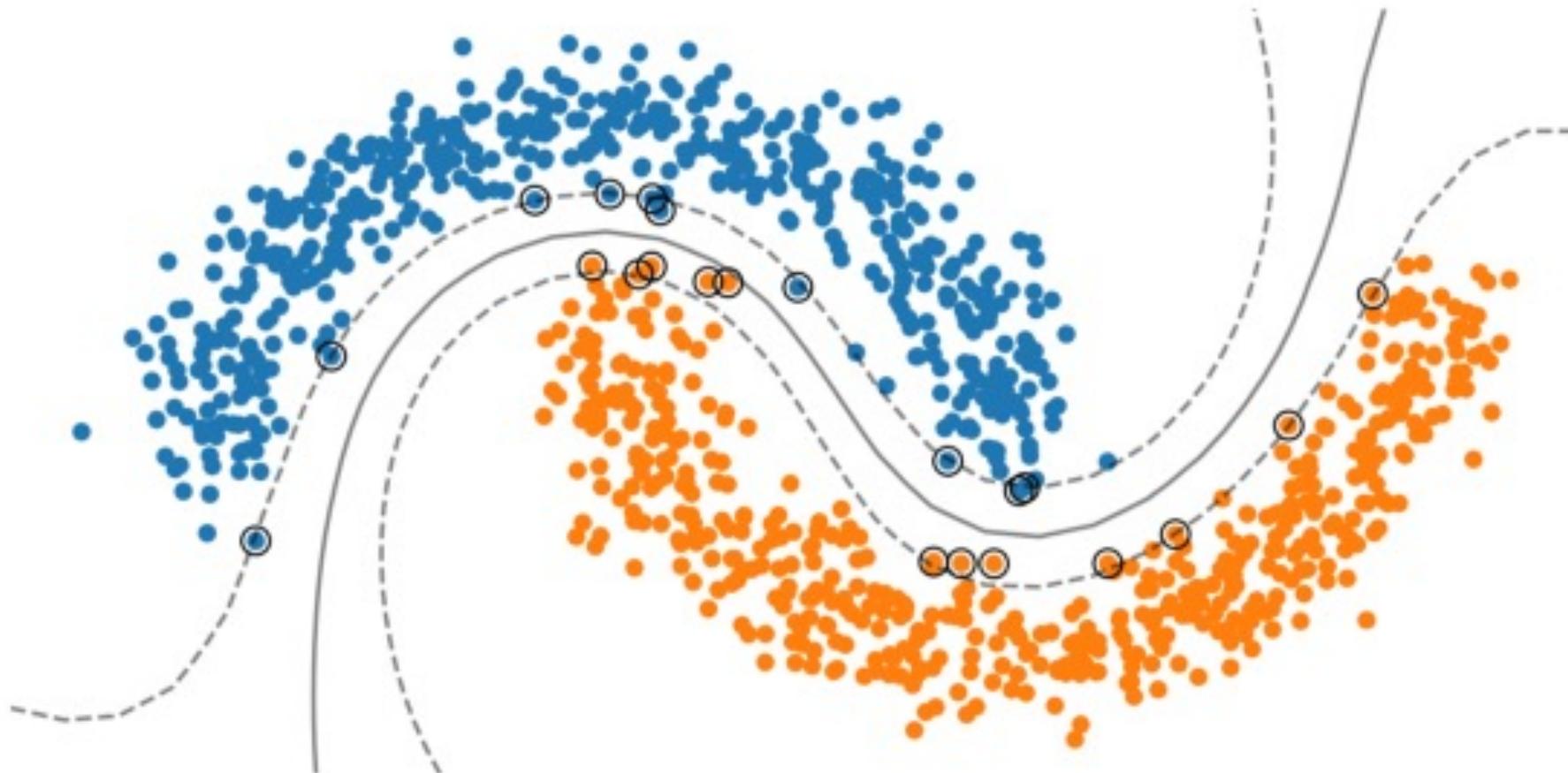
El algoritmo de bosques aleatorios utiliza una versión modificada de *bagging*, en la que para cada árbol de decisión, además de un subconjunto de datos distinto, se utilizan subconjuntos aleatorios de características con el objetivo de construir árboles distintos y no correlacionados.

Esquema de la clasificación usando un Bosque Aleatorio



6

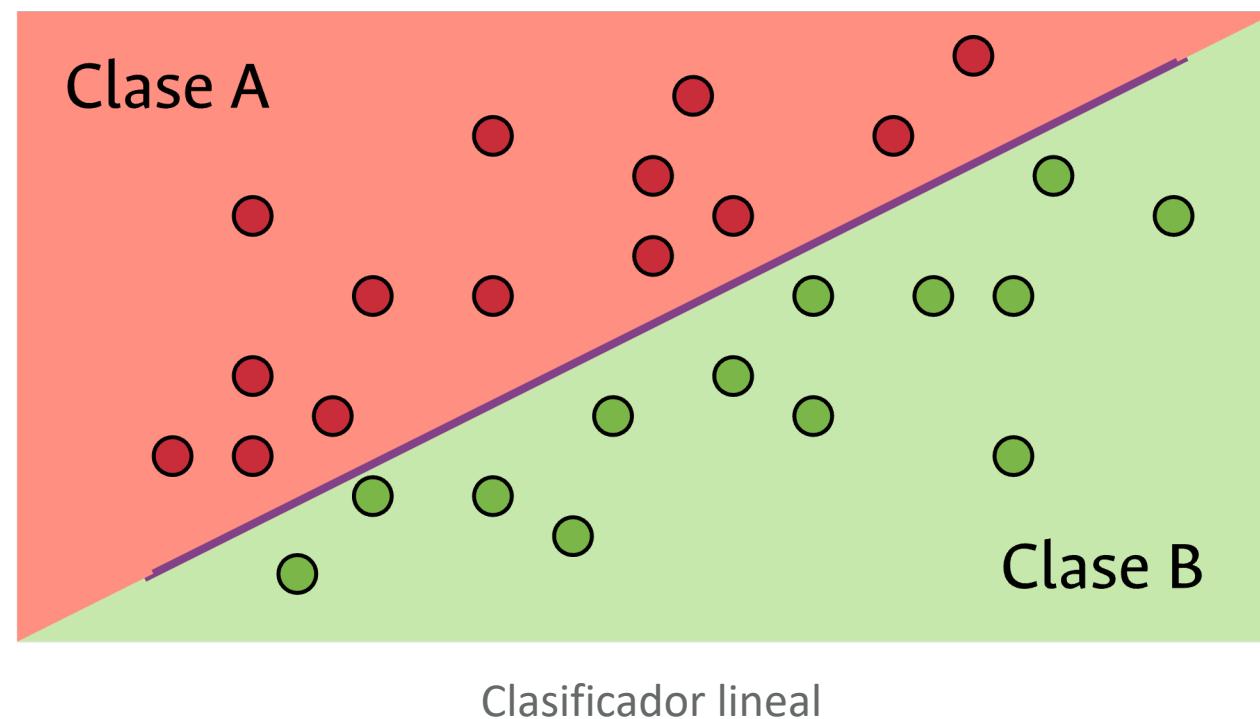
Máquina de Vectores de Soporte (SVM)



SVM

 Clasificador Lineal

Las máquinas de vectores de soporte son un clasificador lineal basado en un enfoque geométrico del aprendizaje computacional. Este clasificador lineal distingue entre dos clases con una frontera lineal.



SVM

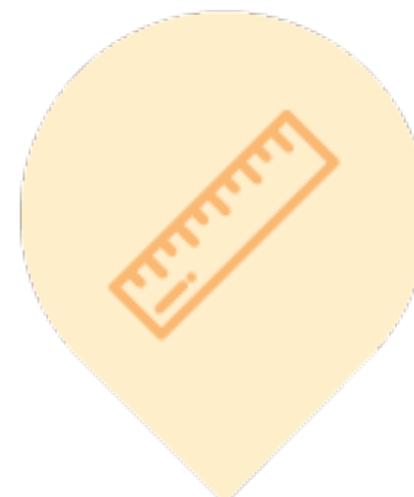


Hiperplano



En geometría, un **hiperplano** es una extensión del concepto de línea a espacios de varias dimensiones, que divide el espacio en dos mitades (Hiperplano, 2019).

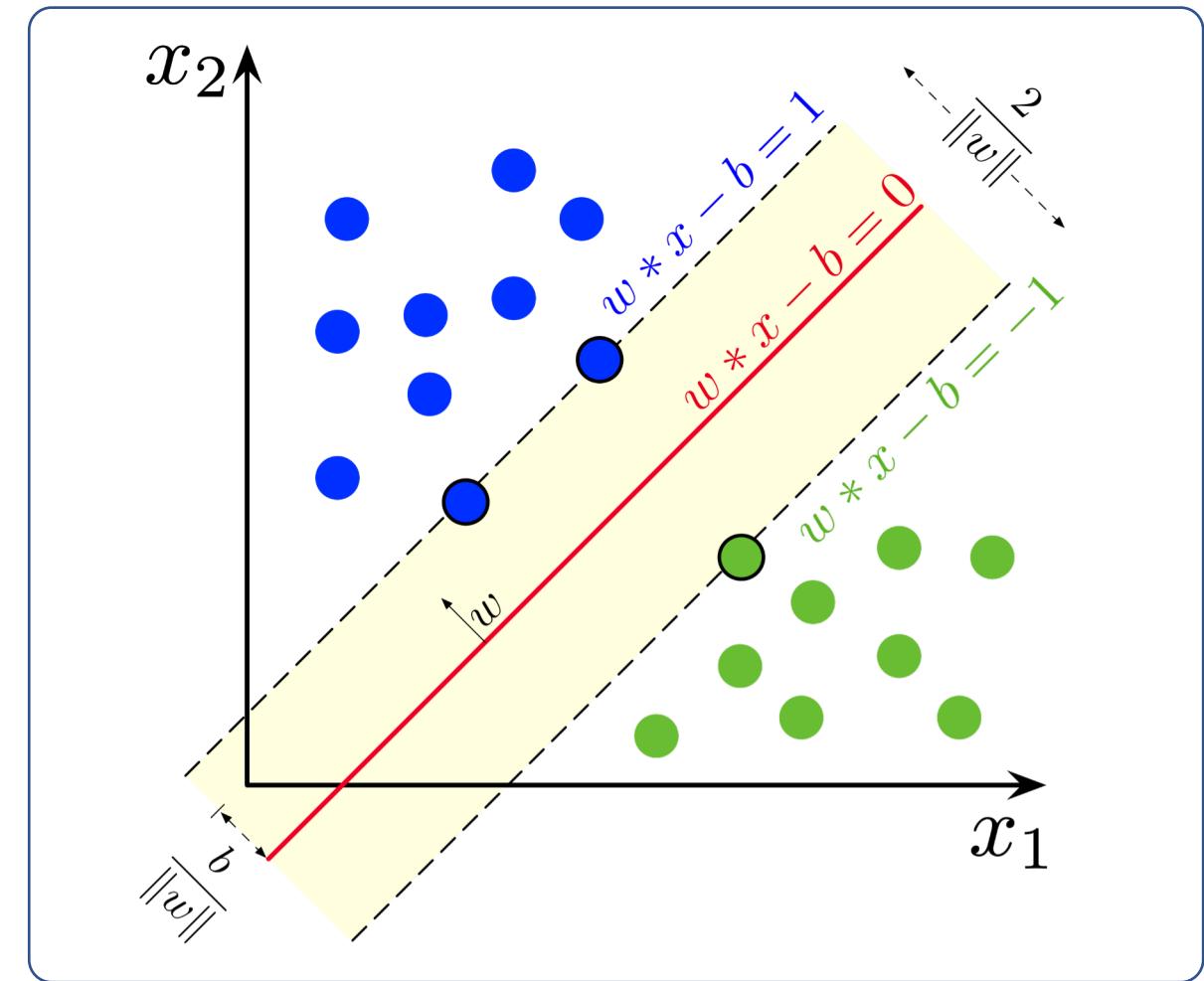
Margen



El margen de un clasificador lineal es la distancia mínima entre el hiperplano y los ejemplos del conjunto de entrenamiento.

SVM

Las máquinas de vectores de soporte encuentran el hiperplano que maximiza el margen entre dos clases. Los vectores (ejemplos) que se encuentran sobre el margen son los vectores de soporte.



SVM

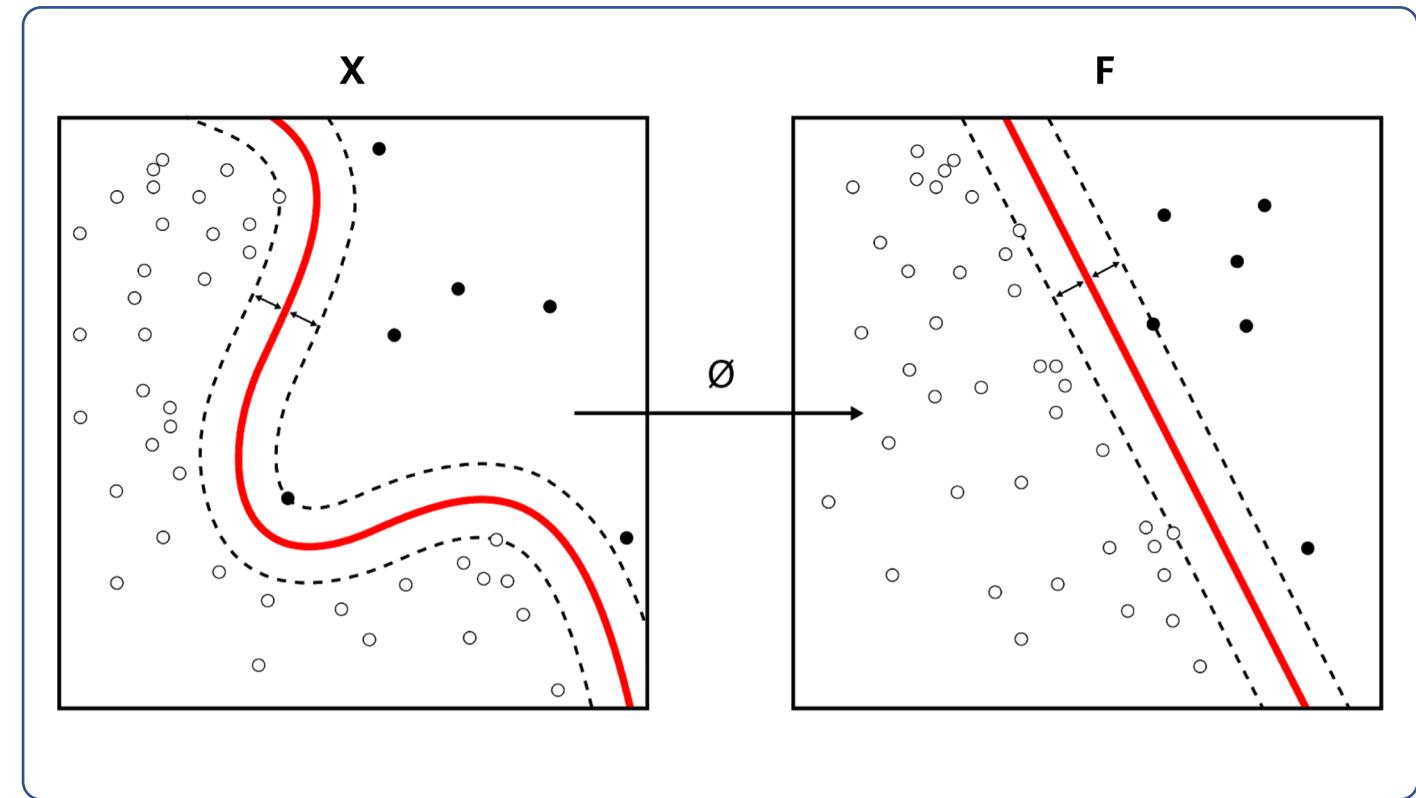


Las máquinas de vectores de soporte pueden ser aplicadas a problemas de clasificación no lineal.

Esto se logra mediante la definición de **un espacio de características (F)**.

En este espacio de característica (F) las relaciones no lineales del espacio de original (X) se vuelven lineales.

La función que transforma los datos del espacio original al espacio de características se define de manera implícita a través de una **función de kernel**.



SVM



Función de Kernel

Función de Kernel

Una función de Kernel K es una función que recibe dos ejemplos y retorna un número real correspondiente a un producto punto en el espacio de características.

$$K: \mathbf{X} \times \mathbf{X} \rightarrow \mathbb{R}$$

$$\mathbf{x}, \mathbf{y} \mapsto \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

Kernel Lineal

$$K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$$

Kernel Polinomial

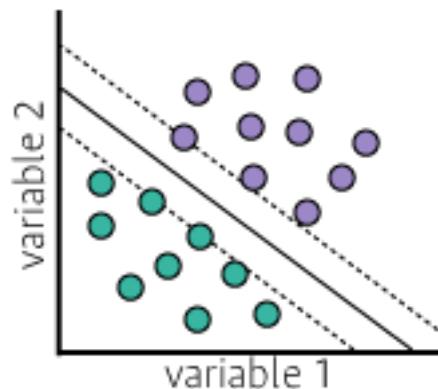
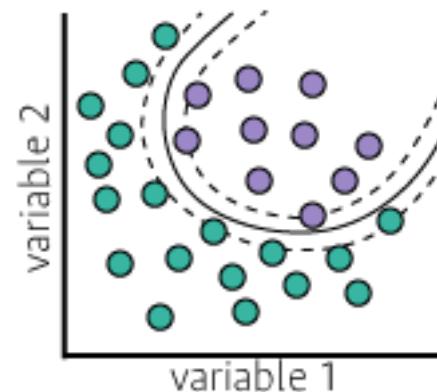
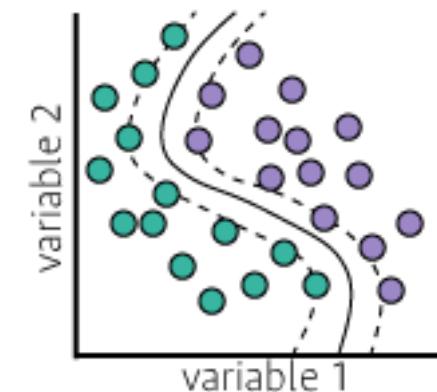
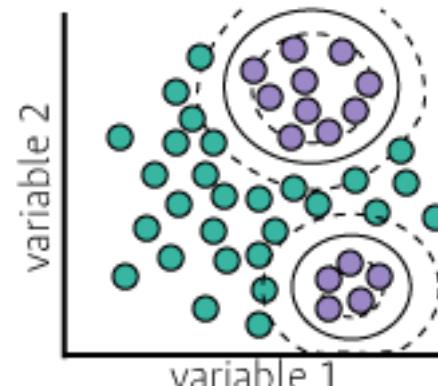
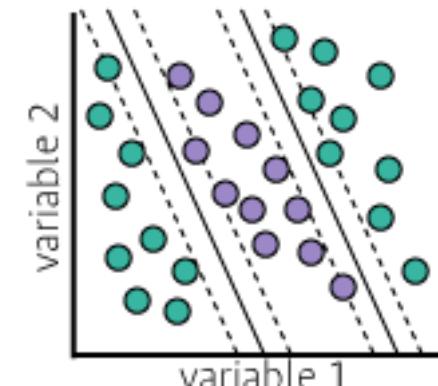
$$K(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$$

Kernel Gaussiano

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)$$

SVM

Función de Kernel

Lineal**Polinomial Grado 2****Polinomial Grado 3****Gaussiano****Gaussiano**

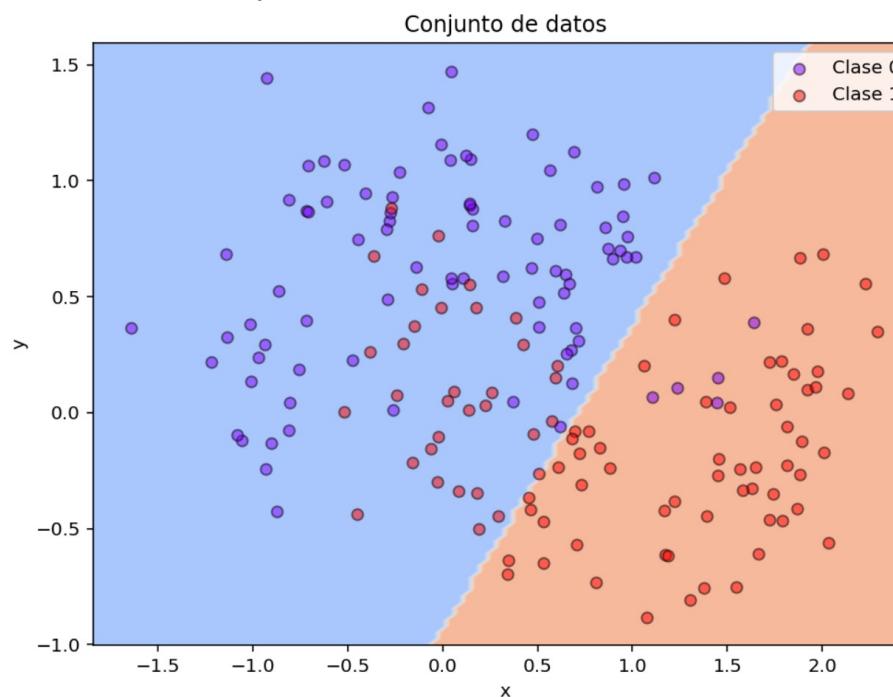
SVM

Función de Kernel

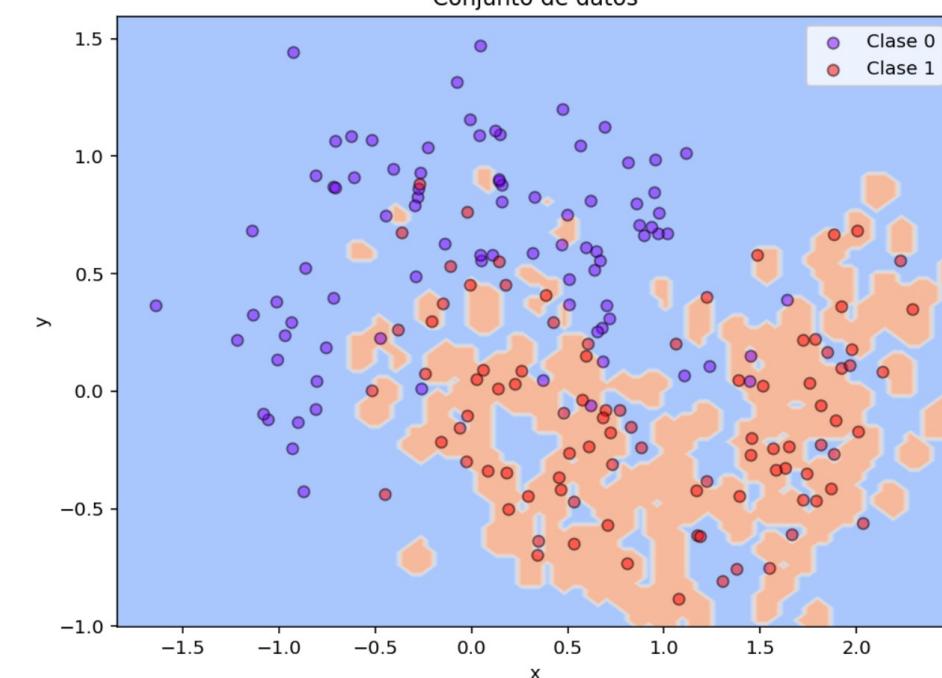
$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right)$$

$$\gamma = \frac{1}{\sigma^2}$$

$$\gamma = 0.0001 \quad \sigma = 100$$



$$\gamma = 1000 \quad \sigma = 0.31$$

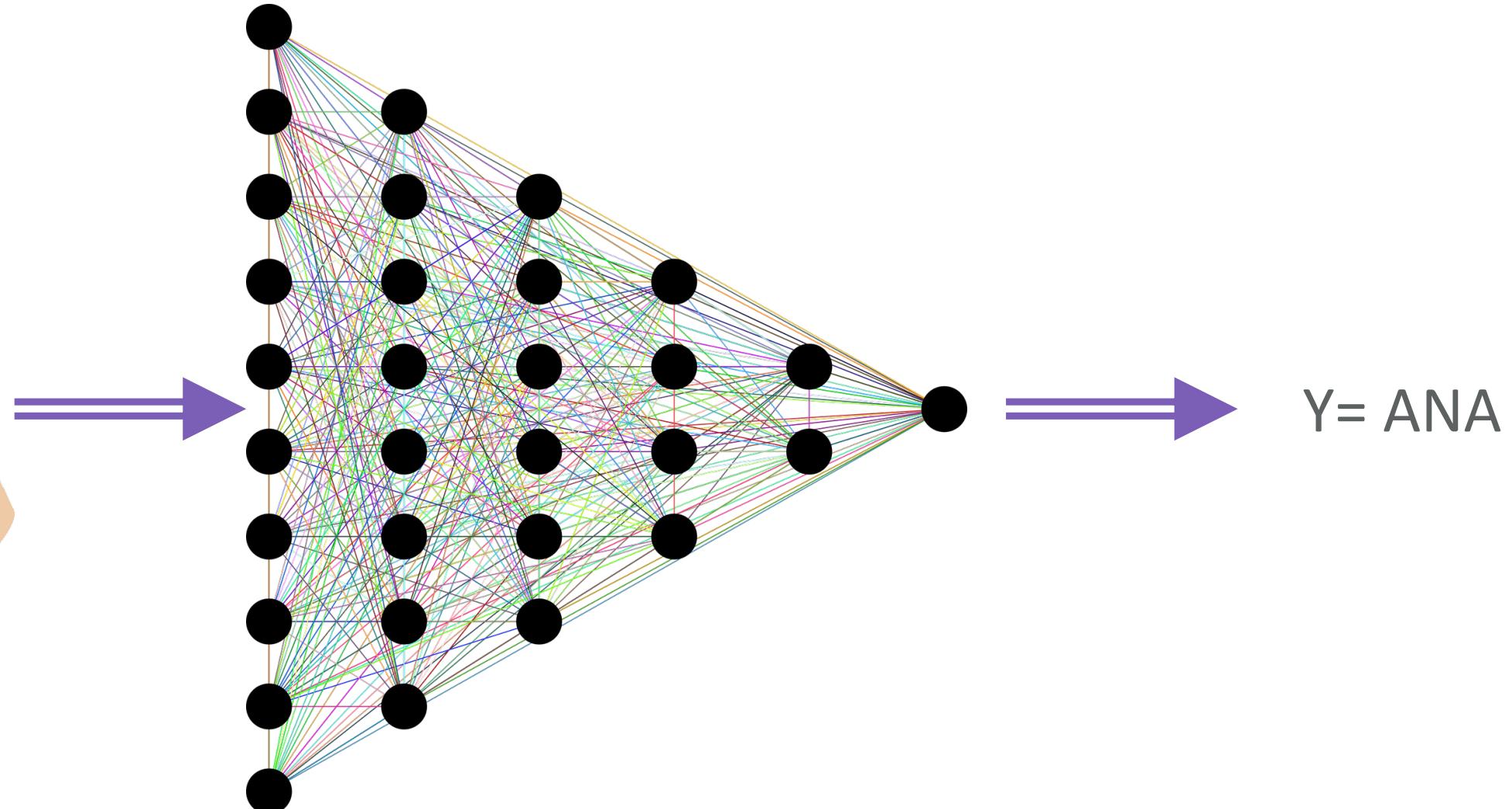


7

Redes Neuronales



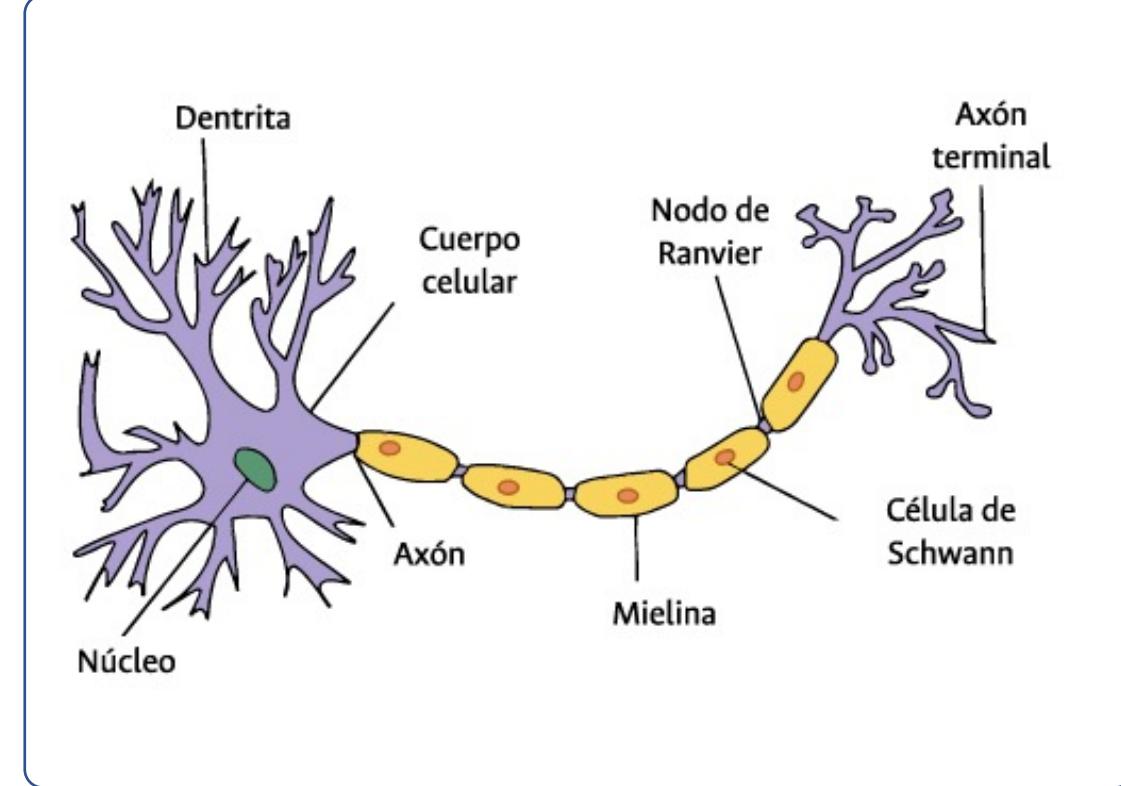
X



Redes Neuronales

Neurona Biológica

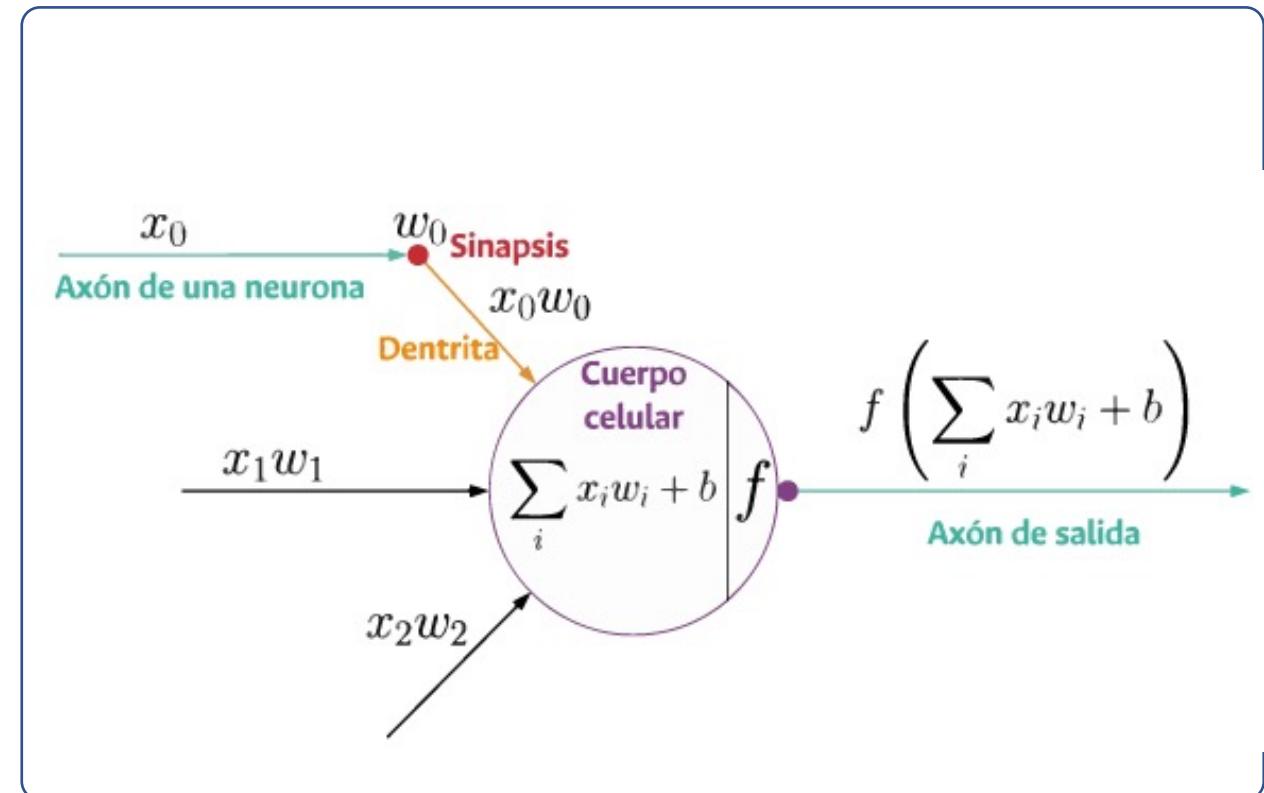
- Aproximadamente, 86 billones de neuronas pueden ser encontradas en el sistema nervioso humano.
- Una neurona biológica está compuesta de:
 - *Dendritas* (entradas)
 - *Núcleo*
 - *Axón* (salidas)
- Los terminales del axón se conectan con las dendritas de otras neuronas a través de conexiones llamadas *sinapsis*.



Redes Neuronales

Neurona Artificial

- Una neurona artificial es un modelo matemático simplificado de la neurona biológica.
- Cada neurona recibe entradas de otras neuronas o por fuentes externas y computa una salida (activación) de la siguiente manera:
 - Cada entrada x_i es multiplicada con un peso w_i , el producto $x_i * w_i$ inhibe(-) o incrementa la activación(+) de la neurona.
 - Se calcula la suma de los productos y un término constante, b , llamado sesgo (bias)
 - Se aplica una función de activación, $f()$, a la suma para obtener la salida

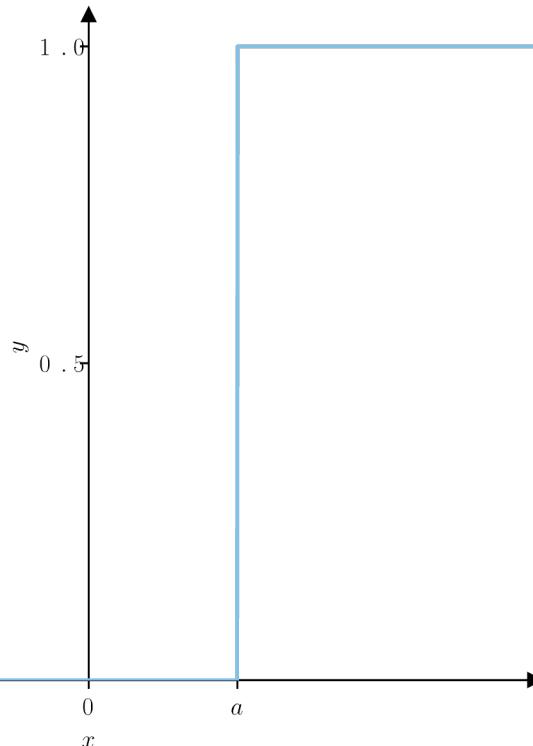


Función Escalón de *Heaviside*:

La función de activación escalón se define como:

$$f(x) = \begin{cases} 1 & : x > a \\ 0 & : x < a \end{cases}$$

Esta función actúa como un umbral en donde los valores mayores que un parámetro dado son mapeados a 1 y el resto a 0 .

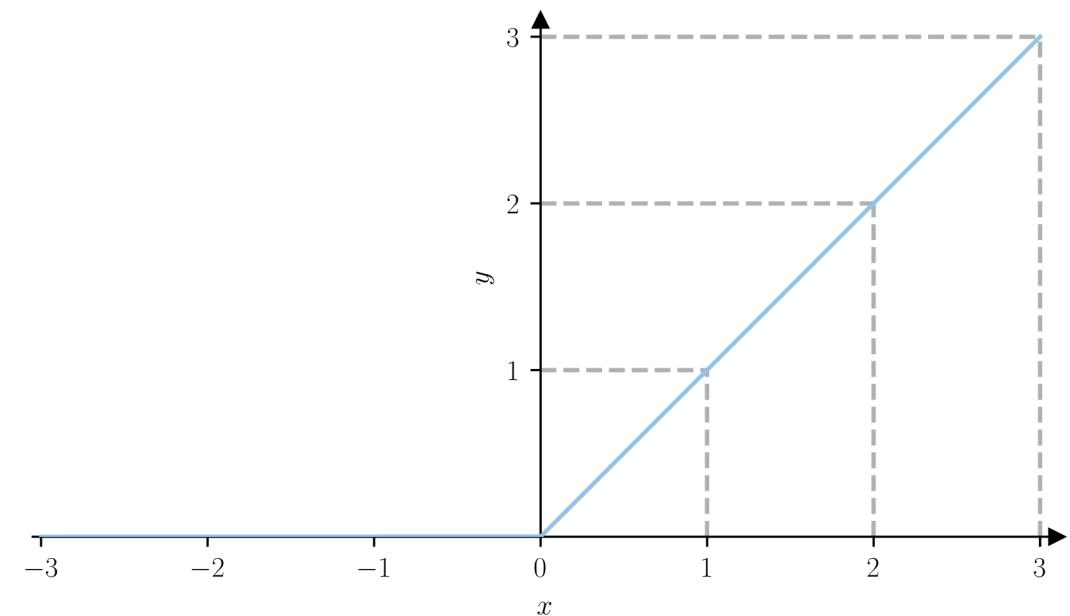


Rectificador Lineal

(ReLU, por sus siglas en inglés)

El *rectificador* es una función de activación definida como:

$$f(x) = \max(x, 0)$$



Redes Neuronales

Funciones de Activación

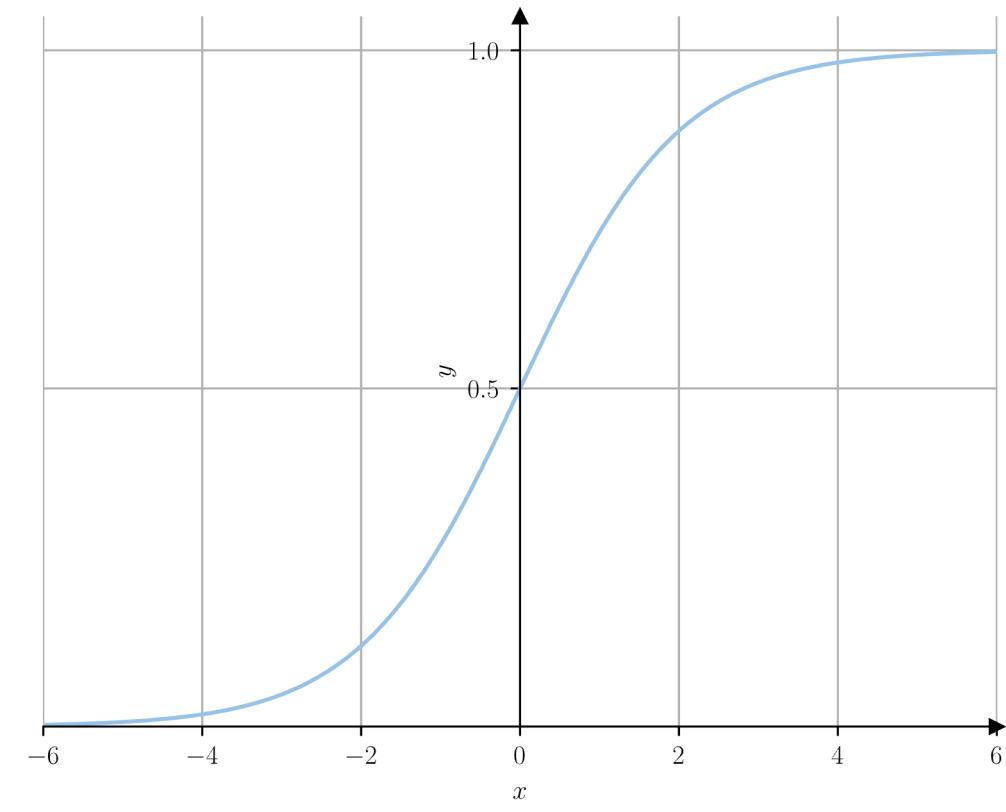
Sigmoide Logística

La función de activación sigmoide se define como:

$$f(x) = \frac{1}{1 + e^{-x}}$$

La función logística mapea números reales al rango $[0, 1]$, en el que se asemeja la probabilidad de un evento; por ejemplo, la probabilidad de la entrada de pertenecer a una clase.

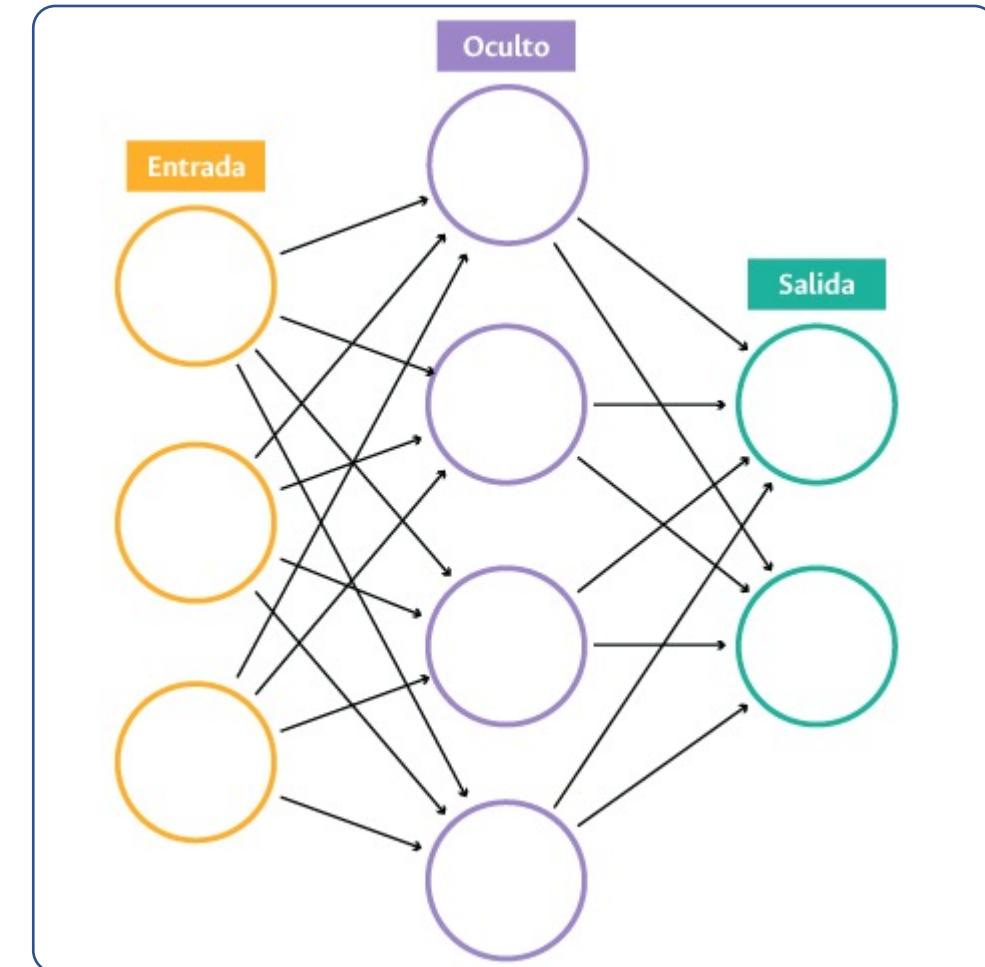
La función logística se emplea en problemas de clasificación binaria.



Redes Neuronales

Arquitectura

- Una red neuronal artificial es un conjunto de neuronas artificiales conectadas de acuerdo con cierto patrón (arquitectura)
- Generalmente están organizadas en capas:
 - **Capa de entrada:** En esta capa no se ejecuta ningún cálculo, simplemente, pasan información a la capa siguiente.
 - **Capas ocultas:** Transfieren las activaciones de la capa de entrada a la siguiente, ya sea otra capa oculta o la capa de salida.
 - **Capa de salida:** En esta capa se utiliza una función de activación que mapea el formato de salida deseado.

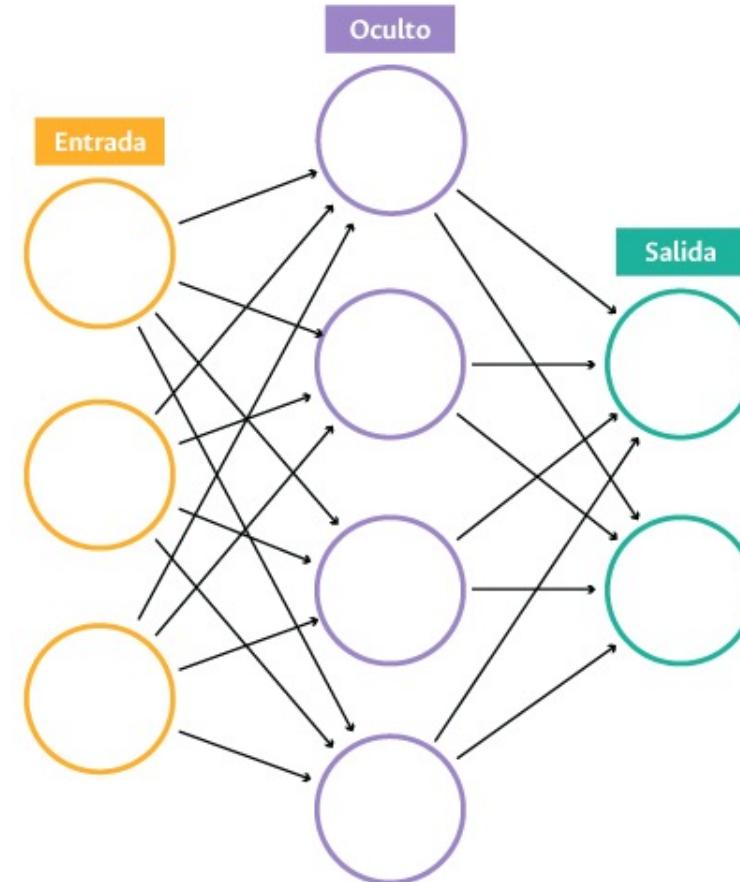


Redes Neuronales

Entrenamiento

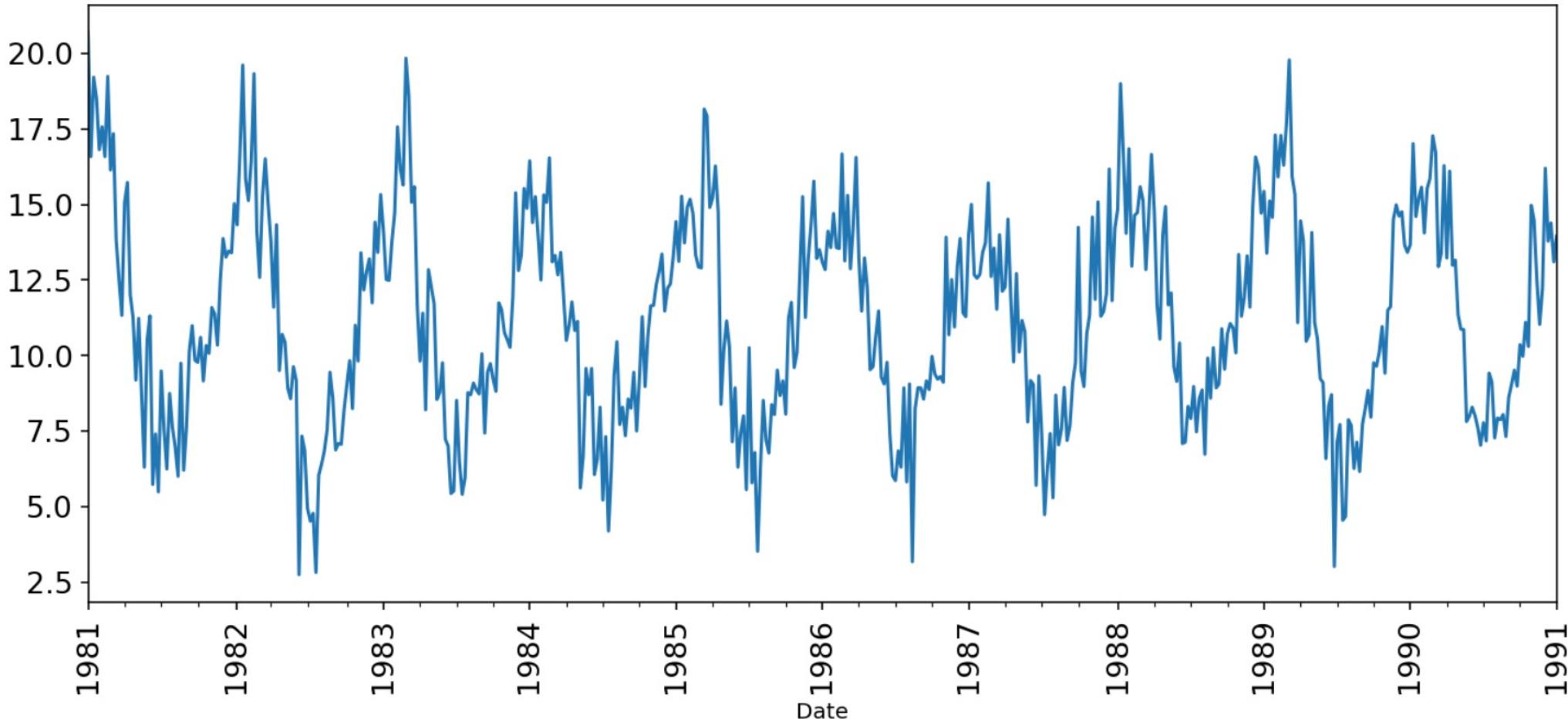
- Los parámetros de una red neuronal son los pesos y sesgo (bias) de cada neurona.
- Actualizar estos parámetros durante el entrenamiento requiere calcular el gradiente de la función de error respecto a estos parámetros.
- El gradiente se calcula con un algoritmo conocido como **Backpropagation** (propagación hacia atrás).
- Los gradientes son calculados primero en la capa de salida y luego capa por capa hacia atrás hasta la capa de entrada.

Arquitectura por capas de una red neuronal artificial



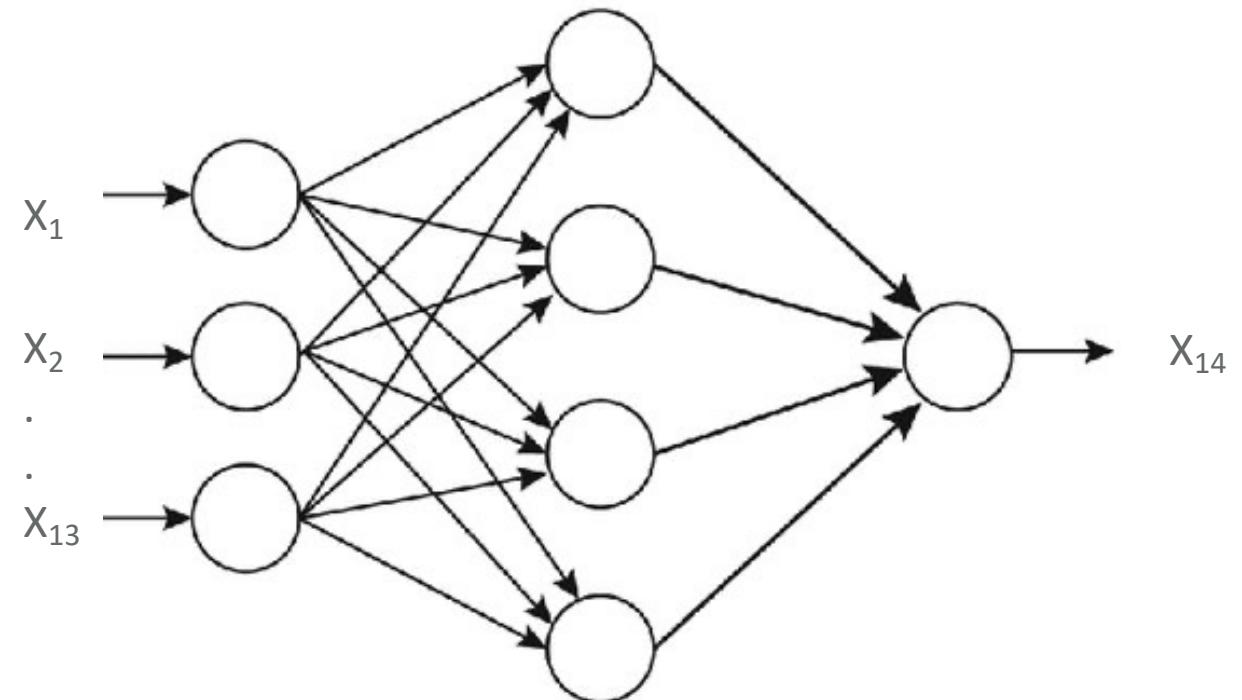
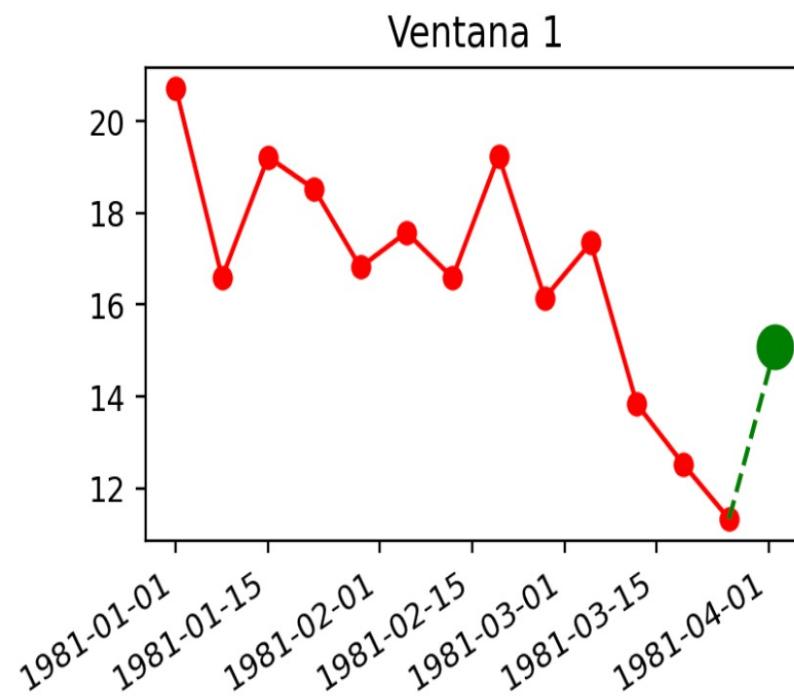
8

Series de Tiempo



Series de tiempo

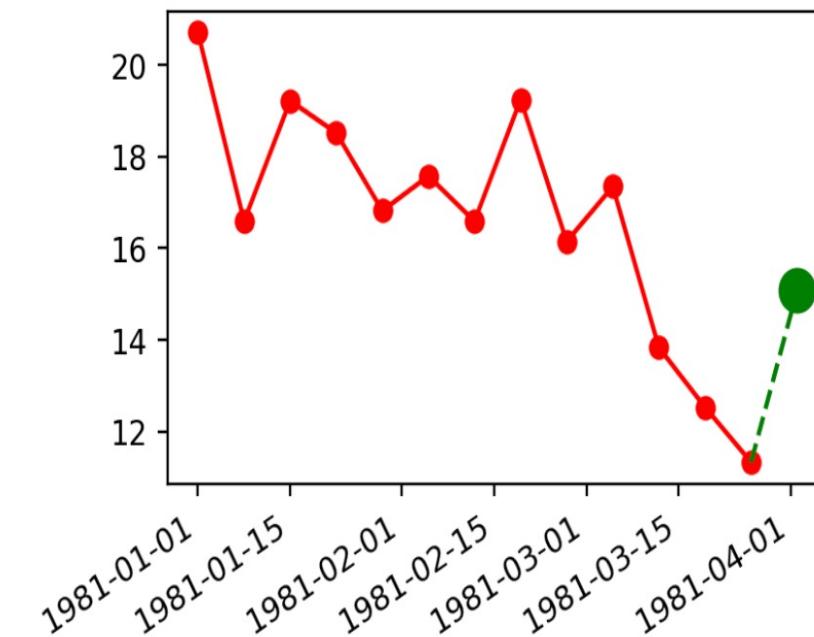
Modelos autoregresivos con redes neuronales



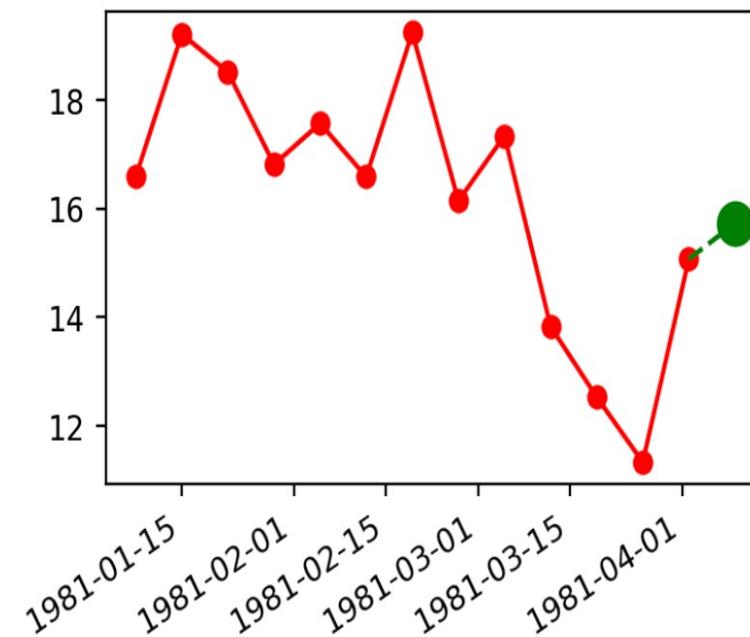
Series de tiempo

Ventanas deslizantes

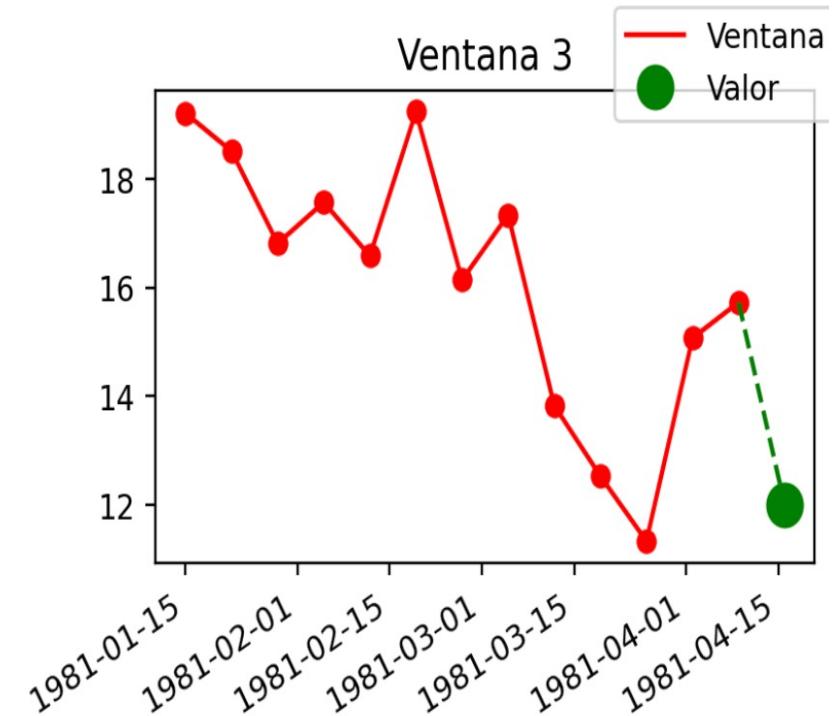
Ventana 1



Ventana 2



Ventana 3



Series de tiempo

 Ventanas deslizantes

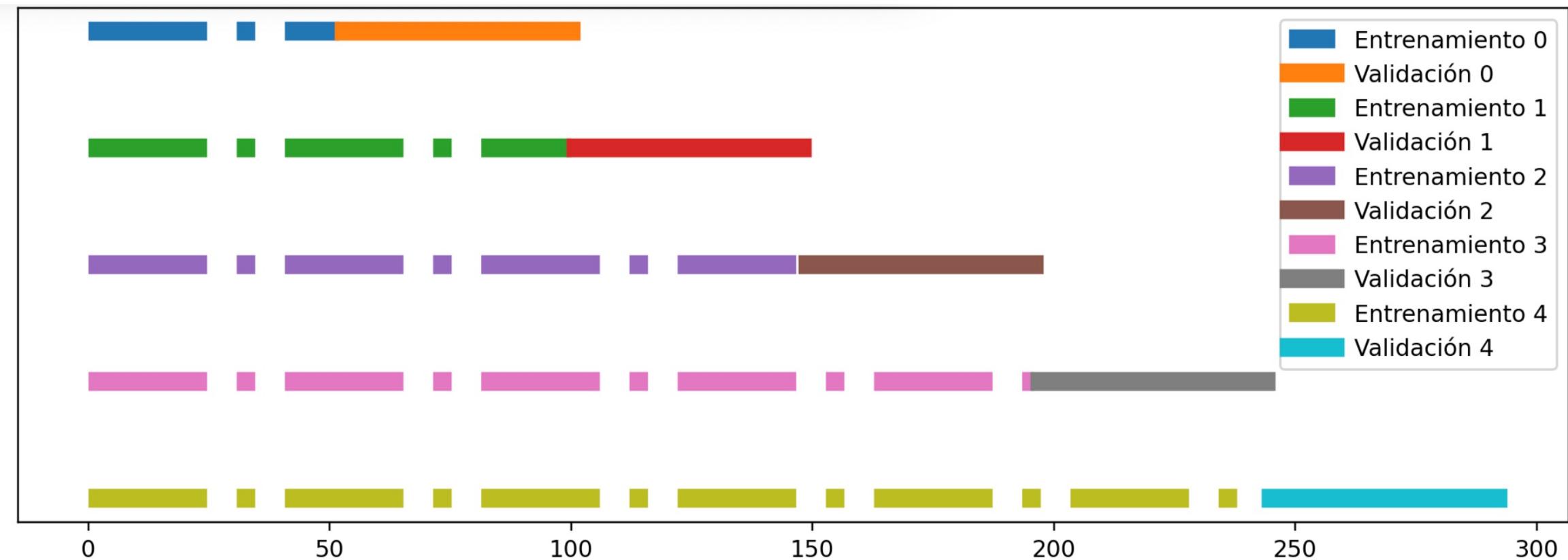
```
1 #Creamos las ventanas y sus valores a predecir para entrenamiento y validación.  
2 x_train, y_train = sliding_time(data_train.values, window_size=k)
```

	0	1	2	3	4	5
0	20.700000	16.585714	19.214286	18.514286	16.814286	17.571429
1	16.585714	19.214286	18.514286	16.814286	17.571429	16.585714
2	19.214286	18.514286	16.814286	17.571429	16.585714	19.242857
3	18.514286	16.814286	17.571429	16.585714	19.242857	16.142857
4	16.814286	17.571429	16.585714	19.242857	16.142857	17.342857
...

Series de tiempo

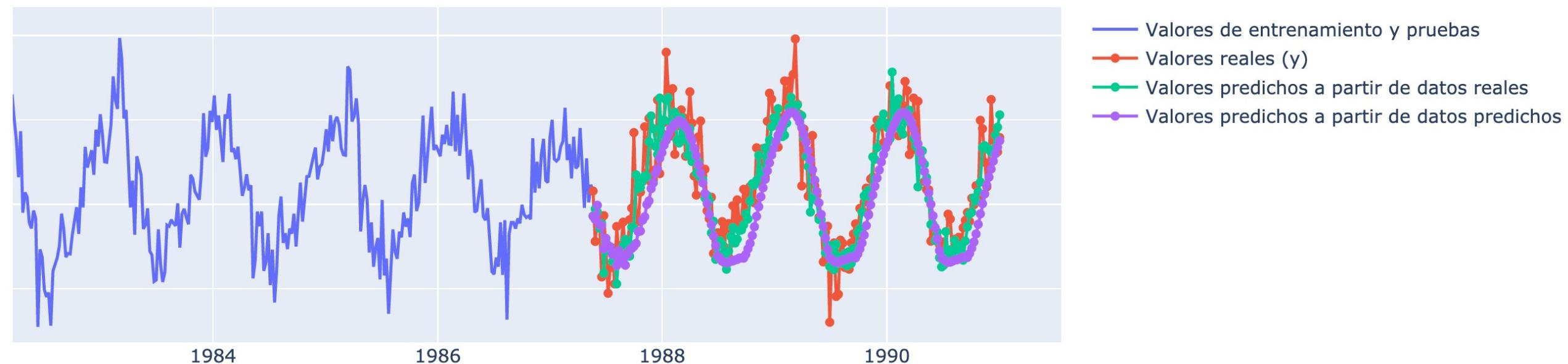
Validación cruzada

```
1 # Definimos el número de splits para realizar cross-validation
2 tsp = TimeSeriesSplit(n_splits=5)
```



Series de tiempo

Predicción





Despedida

Fabio Augusto Gonzalez, PhD.

<https://dis.unal.edu.co/~fgonza/>

fagonzalezo@unal.edu.co



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Departamento de Ingeniería de Sistemas e Industrial
Facultad de Ingeniería
Universidad Nacional de Colombia
Sede Bogotá



Referencias

Función logística. (Junio 20 del 2020). Función Sigmoide Logística estándar [Gráfica].

En Wikipedia. https://en.wikipedia.org/wiki/Logistic_function

Hiperplano. (Agosto 30 del 2019).

En Wikipedia

<https://es.wikipedia.org/wiki/Hiperplano#:~:text=%20geometr%C3%ADa%2C%20un%20hiperplano%20es,una%20l%C3%ADnea%20en%20dos%20l%C3%ADneas>

Hiperplano de margen máximo y márgenes para un SVM entrenado con muestras de dos clases. (Junio 21 del 2020) [Diagrama]. En Wikipedia https://en.wikipedia.org/wiki/Support_vector_machine

Hefinioanrhvs. (Octubre 10 del 2019). Ejemplos con distintas funciones de kernel. [Gráfica]. <https://www.r-bloggers.com/support-vector-machines-with-the-mlr-package/>

Kashnitsky, Y. (s.f.). Árbol de decisión para la otorgación de préstamos [Diagrama de flujo]. <https://mlcourse.ai/articles/topic3-dt-knn/#2.-Decision-Tree>

Koehrsen, W. (27 de diciembre del 2017). Esquema de la clasificación usando un Bosque Aleatorio [Esquema].
<https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>



Referencias

Máquina de vectores soporte. (Junio 21 del 2020) [Plano].

En Wikipedia. https://en.wikipedia.org/wiki/Support_vector_machine

Radchenko V. & Kashnitsky, Y. (s.f.). Esquema del algoritmo de ensambles Empaquetado (Bagging) [Imagen].

<https://mlcourse.ai/articles/topic5-part1-bagging/>

Radchenko V. & Kashnitsky, Y. (s.f.). Esquema del algoritmo de ensambles Empaquetado (Bagging) [Esquema].

<https://mlcourse.ai/articles/topic5-part1-bagging/>

Sayad, S. (s.f.). Conjunto de datos Manejar Bicicleta [Tabla].

<https://www.saedsayad.com/classification.htm>

Tan, P., Steinbach, M. & Kuman, V. (Abril 18 del 2004). Ejemplo del vecino más cercano [Diagrama].

https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap5_alternative_classification.pdf

Tan, P., Steinbach, M. & Kuman, V. (Abril 18 del 2004). *Idea básica clasificador KNN* [Diagrama].

https://en.wikipedia.org/wiki/Support_vector_machine

Tan, P., Steinbach, M. & Kuman, V. (Abril 18 del 2004). Ejemplo proceso de predicción clasificador knn [Diagrama].

https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap5_alternative_classification.pdf



Recursos adicionales

Caparrini, F. (2019). Redes Neuronales: una visión superficial.
<http://www.cs.us.es/~fsancho/?e=72>

Kashnitsky, Y (s.f.). Tema 3. Clasificación, Árboles de decisión y Vecinos más cercanos. Curso de Open Machine Learning
<https://mlcourse.ai/articles/topic3-dt-knn/#2.-Decision-Tree>

Kashnitsky, Y (s.f.). Tema 5. Ensambls. Parte 2 Bosques Aleatorios. Curso de Open Machine Learning
<https://mlcourse.ai/articles/topic5-part2-rf/>

Knut Hinkelmann. (s.f.). Neural Networks [presentación de diapositivas]
http://didattica.cs.unicam.it/lib/exe/fetch.php?media=didattica:magistrale:kebi:ay_1718:ke-11_neural_networks.pdf

Sanderson, G. (2018). Redes Neuronales. 3Blue1Brown Youtube
https://www.youtube.com/playlist?list=PLZHQBObOWTQDNU6R1_67000Dx_ZCJB-3pi

Sayad, S. (s.f.). Clasificación, Una introducción a la ciencia de datos.
<https://www.saedsayad.com/classification.htm>

Torres Barrán, A. (2019). Support Vector Machines. Curso de aprendizaje automático para el INE
<http://albertotb.com/curso-ml-R/Rmd/07-svm/07-svm.html#1>



Derechos de imágenes

Creado por [scikit-learn](#):

- Imágenes sección scikit-learn.
- Logo scikit-learn.

<https://pixabay.com/vectors/neural-network-thought-mind-mental-3816319/>

<https://pixabay.com/vectors/student-teenager-book-learning-147783/>

Vintage Tone. (s.f.). Bullseye is a target of business. [Fotografía].

<https://www.shutterstock.com/es/image-photo/bullseye-target-business-dart-opportunity-dartboard-678189055>

Freepik. (s.f.). Search free icon. [Icono].

https://www.flaticon.com/free-icon/search_3409500?term=search%20question&page=1&position=12

Kashnitsky. (s.f.). Árbol de decisión para la otorgación de préstamos. [Gráfico].

Adaptado de <https://mlcourse.ai/articles/topic3-dt-knn/>



Derechos de imágenes

Tan, P., Steinbach, M., Karpatne, A. y Kumar, V. (2004). Árbol de decisión para la otorgación de préstamos. [Gráfico].

Adaptado de <https://www-users.cs.umn.edu/~kumar001/dmbook/index.php>

Kashnitsky. (s.f.). Esquema de algoritmo de ensamble empaquetado-muestreo. [Gráfica].

Adaptado de <https://mlcourse.ai/articles/topic5-part1-bagging/>

Koehrsen. (2017). Esquema predicción bosques aleatorios. [Gráfica].

Adaptado de <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>

Revolucion IA. (2017). Clasificador lineal. [Gráfica].

Adaptado de https://www.revolucionia.com/2017/08/Clasificacion_frente_modelizacion.html

Wikipedia. (2020). Diagrama SVM. [Gráfica].

Adaptada de https://commons.wikimedia.org/wiki/File:SVM_margin.png

Wikipedia. (2020). Espacio de características original y Espacio de características Inducido. [Gráfica].

Adaptada de https://commons.wikimedia.org/wiki/File:Kernel_Machine.svg



Derechos de imágenes

Hefinioanrhvs. (2019). Ejemplos con distintas funciones de Kernel. [Gráfica].

Adaptada de <https://www.r-bloggers.com/support-vector-machines-with-the-mlr-package/>

Wikipedia. (s.f.). Neurona biológica. [Gráfica].

Adaptada de <https://commons.m.wikimedia.org/wiki/File:Neuron.svg>

Stanford CS231n. (s.f.). Neurona artificial. [Gráfica].

Adaptado de CS231n Convolutional Neural Networks for Visual Recognition <https://cs231n.github.io/neural-networks-1/>

Wikipedia. (s.f.). Arquitectura por capas de una red neuronal artificial. [Gráfica].

Adaptada de https://es.wikipedia.org/wiki/Red_neuronal_artificial#/



Créditos

Facultad de
INGENIERÍA

Autores

Fabio Augusto González Osorio, PhD

Asistente docente

Miguel Ángel Ortiz Marín

Diseño instruccional

Claudia Patricia Rodríguez Sánchez

Diseño gráfico

Clara Valeria Suárez Caballero

Milton R. Pachón Pinzón

Diagramadora PPT

Daniela Duque

Fecha
2021-I

