

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

Facultad de  
**INGENIERÍA**





# Módulo 2

# Introducción al Machine

# Learning con *Python*

Unidad 2

Desarrollo de modelos de aprendizaje  
computacional

Clase sincrónica

Facultad de  
**INGENIERÍA**



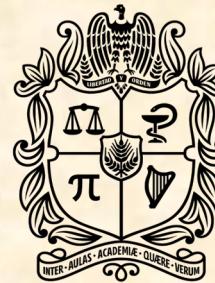


# Bienvenida

## Fabio Augusto Gonzalez, PhD.

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

[fagonzalezo@unal.edu.co](mailto: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 Experimentos de aprendizaje computacional

2 Generalización

- Sobreajuste, subajuste y ajuste apropiado
- Capacidad del modelo
- Parámetros e hiperparámetros

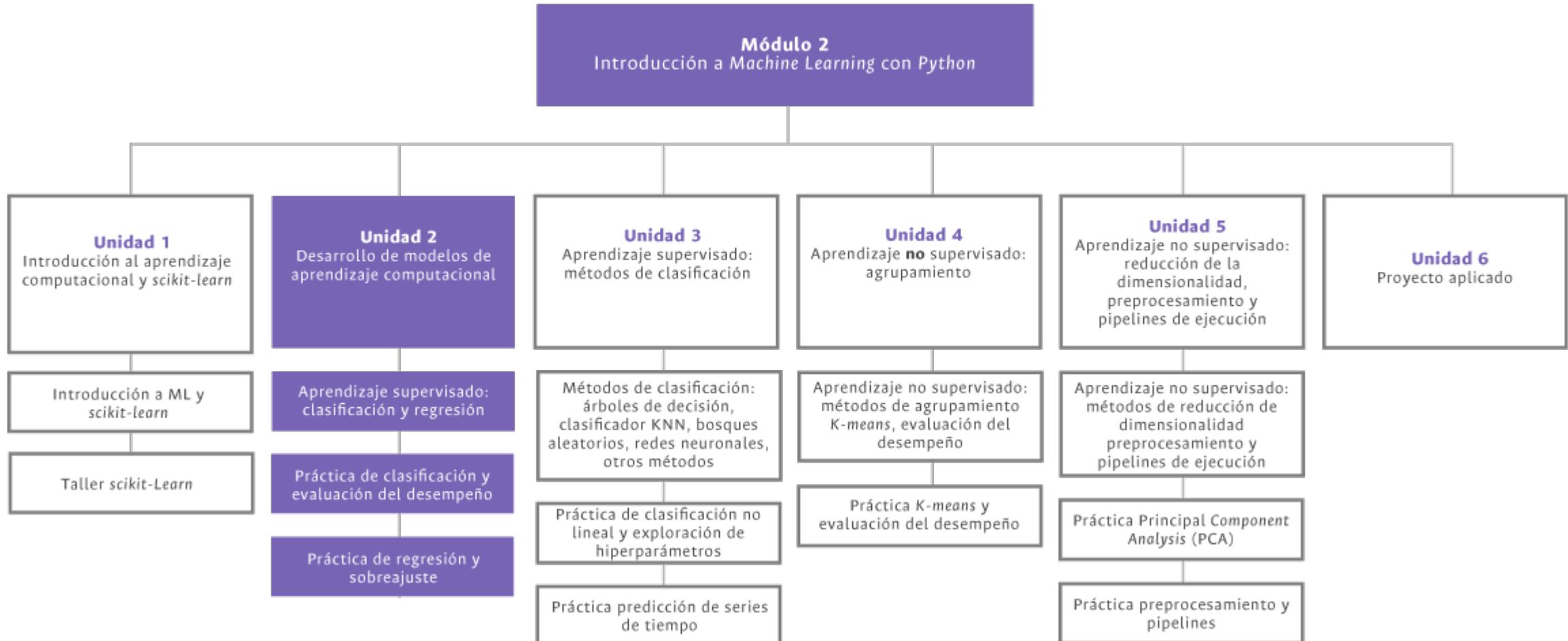
3 Conjunto de entrenamiento, validación, prueba

- Flujo de trabajo
- Validación cruzada.

4 Métricas de desempeño



# Mapa de contenidos de la unidad



## Objetivos de aprendizaje



## Unidad 2 - Desarrollo de modelos de aprendizaje computacional

Al finalizar la unidad usted deberá ser capaz de:

 1

Describir en qué consisten las tareas de clasificación y regresión.

 2

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

 3

Diseñar un experimento de aprendizaje computacional.

## Objetivos de aprendizaje



## Unidad 2 - Desarrollo de modelos de aprendizaje computacional

Al finalizar la unidad usted deberá ser capaz de:



4

Evaluar modelos de clasificación mediante el uso de diferentes métricas de desempeño.

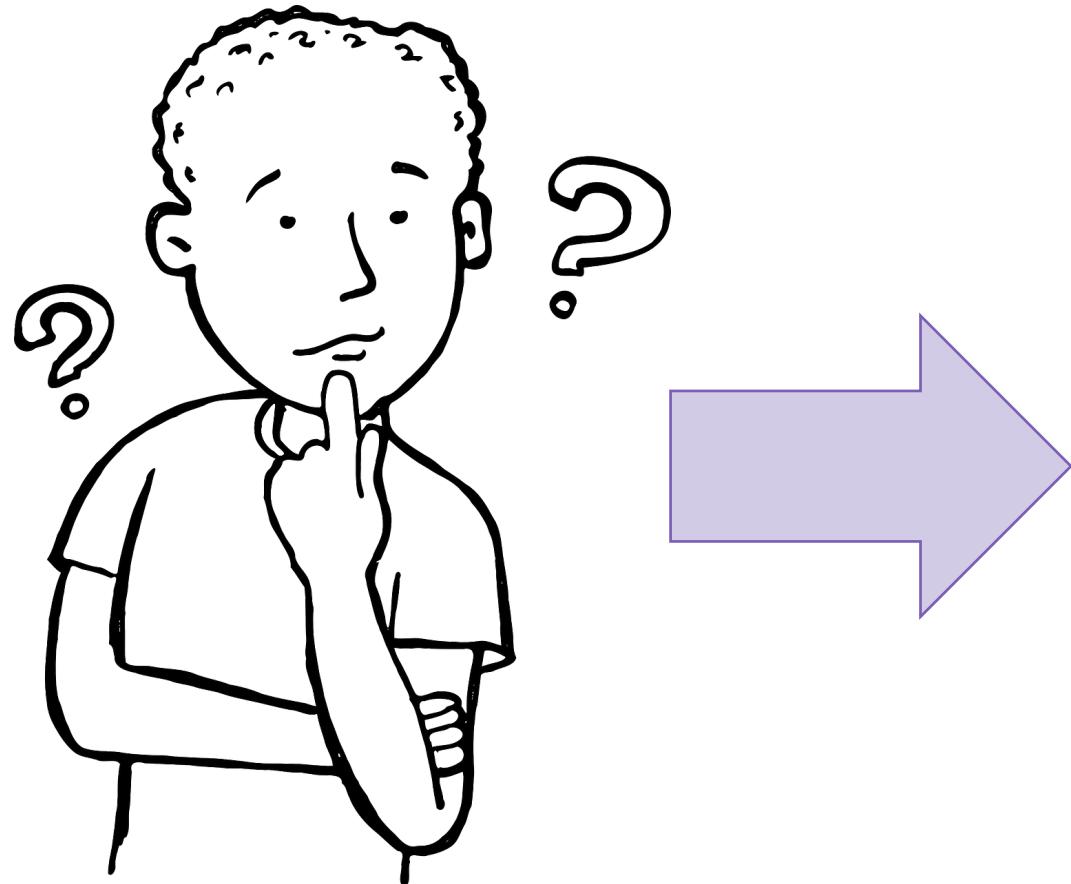


5

Controlar el sobreajuste mediante el uso de estrategias de validación cruzada.

## 1

# Experimentos de aprendizaje computacional



## Experimentos de aprendizaje computacional

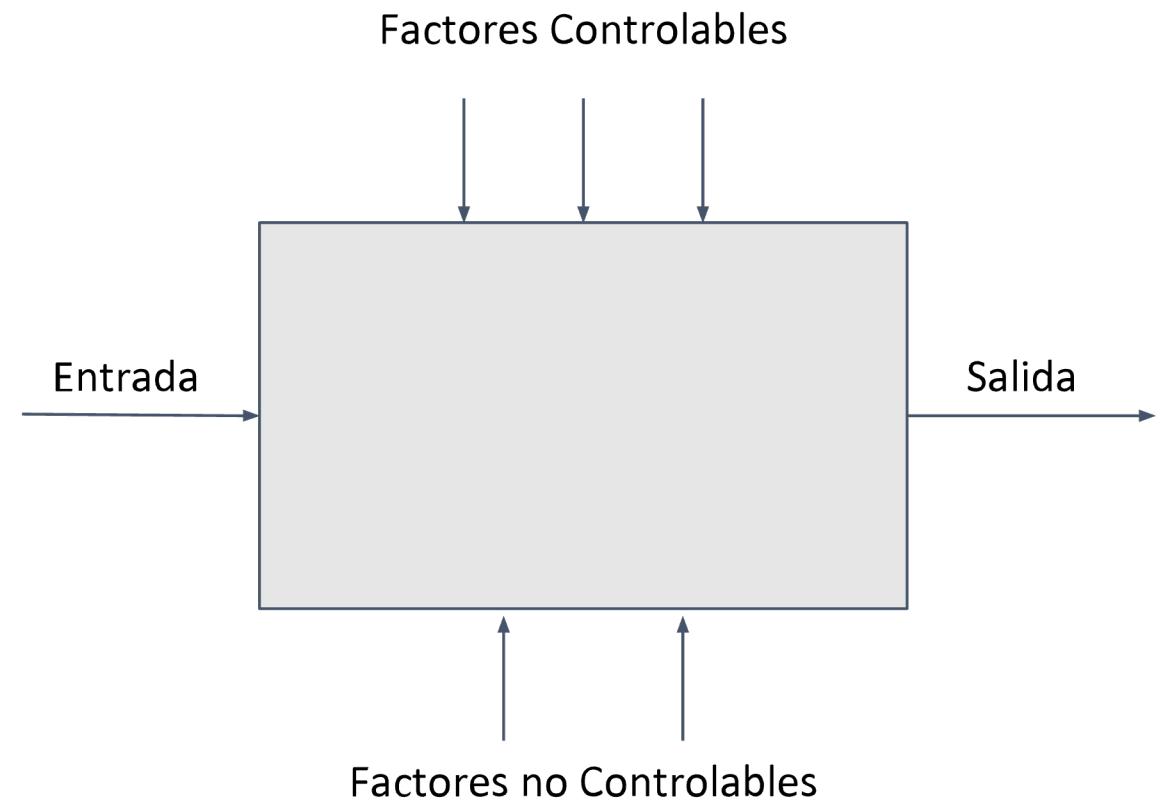
- El desarrollo de sistemas basados en aprendizaje computacional es fundamentalmente un proceso de tipo experimental.
- El proceso experimental evalúa de manera sistemática diferentes modelos para resolver un problema.
- El resultado final de este proceso es un modelo que resuelva de manera satisfactoria el problema de aprendizaje abordado.
- Este proceso difiere del enfoque utilizado para resolver otro tipo de problemas con el uso de computador.
- En el aprendizaje computacional se hace más énfasis en la experimentación que, por ejemplo, en la escritura de código.



## Experimentos de aprendizaje computacional

En el aprendizaje computacional el objeto que se observa bajo la lupa durante los experimentos son los modelos, los cuales, al ser entrenados, producen una salida para una determinada entrada.

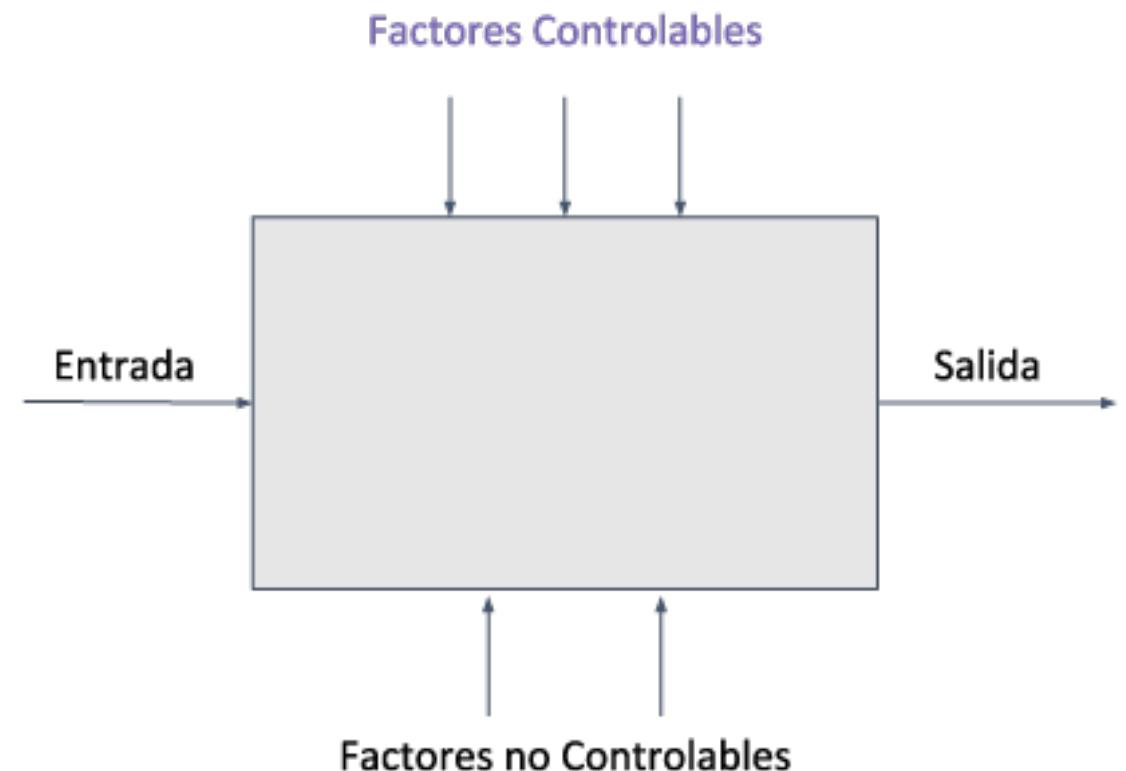
El experimento consiste en una prueba o serie de pruebas en los que se juega con los factores que afectan la salida, entonces, se busca encontrar la configuración de factores que maximicen la respuesta. Por ejemplo, la exactitud de clasificación en un conjunto de datos de prueba.



## Experimentos de aprendizaje computacional

## Factores controlables

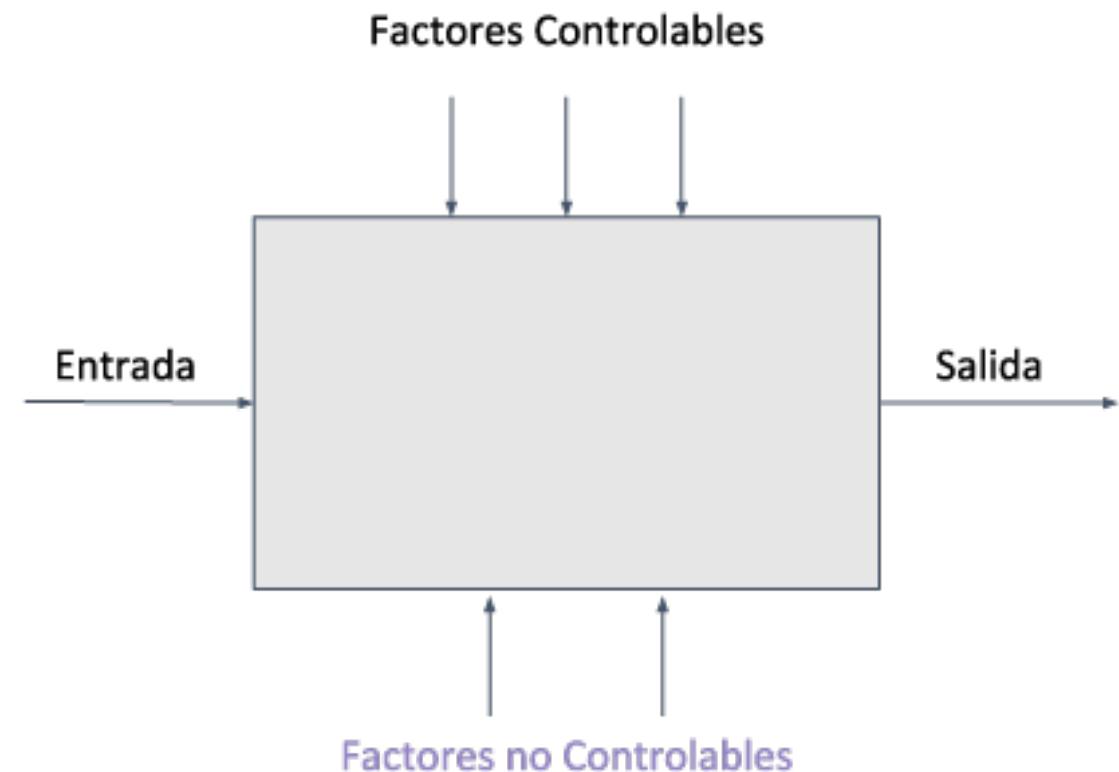
- El algoritmo de aprendizaje utilizado
- Los hiperparámetros del algoritmo
- El subconjunto de datos para entrenar y para probar
- La representación de la entrada, es decir, la manera en que la entrada es codificada.



## Experimentos de aprendizaje computacional

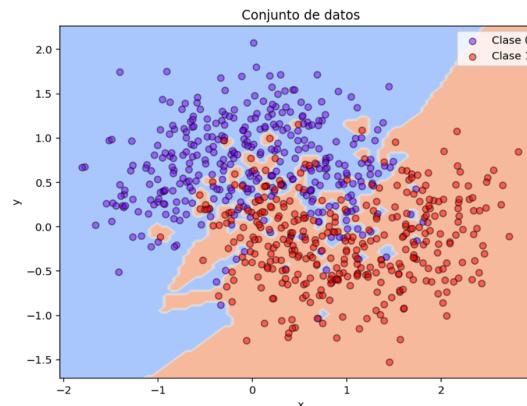
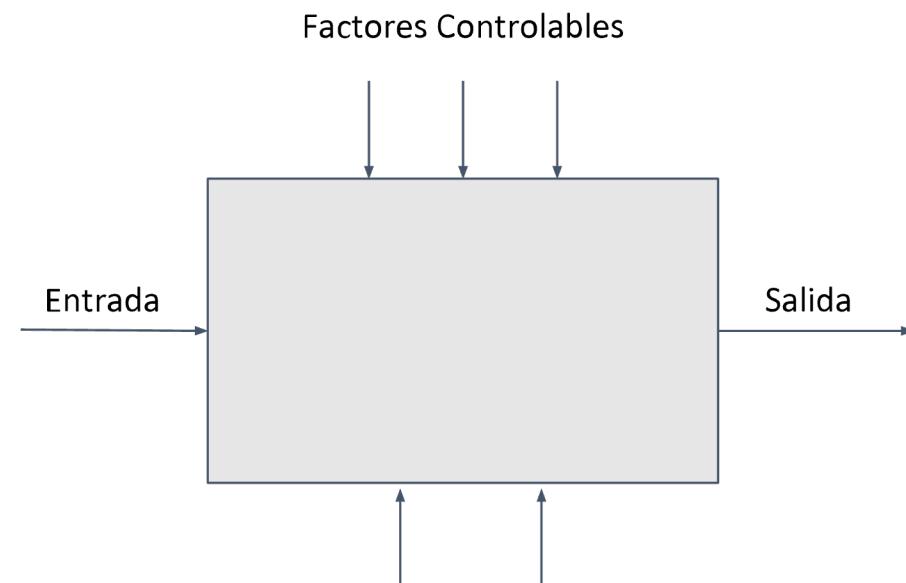
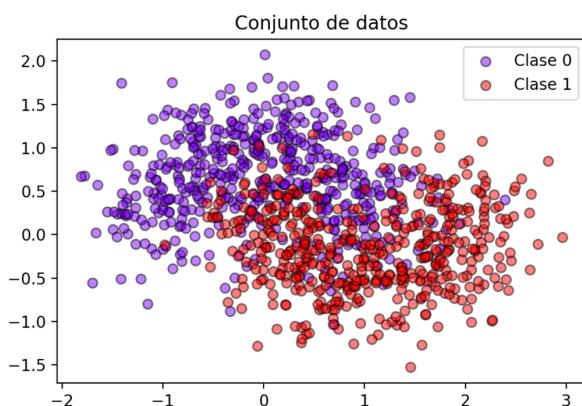
Factores NO controlables

- Ruido de los datos
- Aleatoriedad en el proceso de entrenamiento
- La selección aleatoria del conjunto de entrenamiento



## Experimentos de aprendizaje computacional / Ejemplo

Algoritmo: k-vecinos más cercanos  
 $k: 1$



Ruido en los datos  
Errores en los datos

Error entrenamiento: 0%  
Error de prueba: 20%

## Experimentos de aprendizaje computacional

## Tipos de experimentos

La experimentación se realiza con diferentes objetivos.

Los siguientes son objetivos típicos de la experimentación:



Experimentos para seleccionar características



Experimentos para ajustar un modelo



Experimentos para comprobar modelos

## Experimentos de aprendizaje computacional

## Tipos de experimentos

## Experimentos para seleccionar características



- El objetivo es seleccionar las características que mejor representan los datos de un conjunto de características candidatas.
- Tener un conjunto de buenas características tiene un impacto positivo en la interpretabilidad y eficiencia del modelo.
- El factor controlable corresponde las características seleccionadas.
- La respuesta se evalúa en términos de alguna métrica de desempeño en el conjunto de prueba.

## Experimentos de aprendizaje computacional

## Tipos de experimentos

## Experimentos para ajustar un modelo



- El objetivo es configurar un modelo particular de manera que tenga el mejor desempeño posible.
- El factor controlable corresponde a los hiperparámetros del modelo.
- Los hiperparámetros son las opciones que podemos usar para controlar el comportamiento del algoritmo de aprendizaje. Por ejemplo, el grado de un polinomio en regresión polinomial o la profundidad máxima de un árbol de decisión.
- La respuesta se evalúa en términos de alguna métrica de desempeño en el conjunto de prueba.

## Experimentos de aprendizaje computacional

## Tipos de experimentos

## Experimentos para comparar modelos



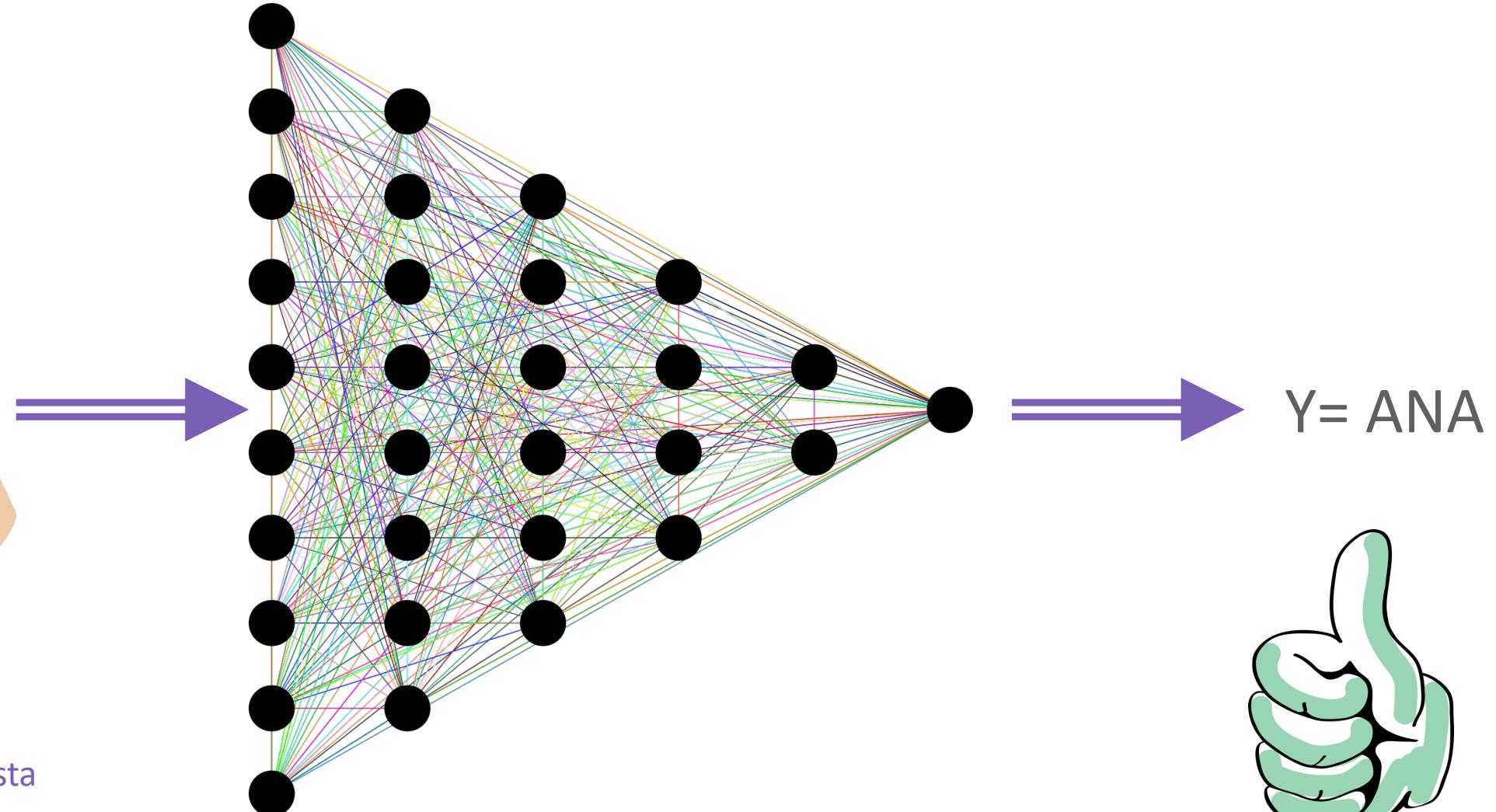
- El objetivo es encontrar el modelo que resuelve de mejor manera un problema particular.
- El factor controlable corresponde al tipo de modelo.
- Hay diferentes tipos de modelos para las diferentes tareas de aprendizaje, por ejemplo: árboles de decisión, redes neuronales, modelos Bayesianos, etc.
- Cada tipo de modelo puede funcionar mejor en diferentes tipos de problemas.
- La respuesta se evalúa en términos de alguna métrica de desempeño en el conjunto de prueba.

## Generalización



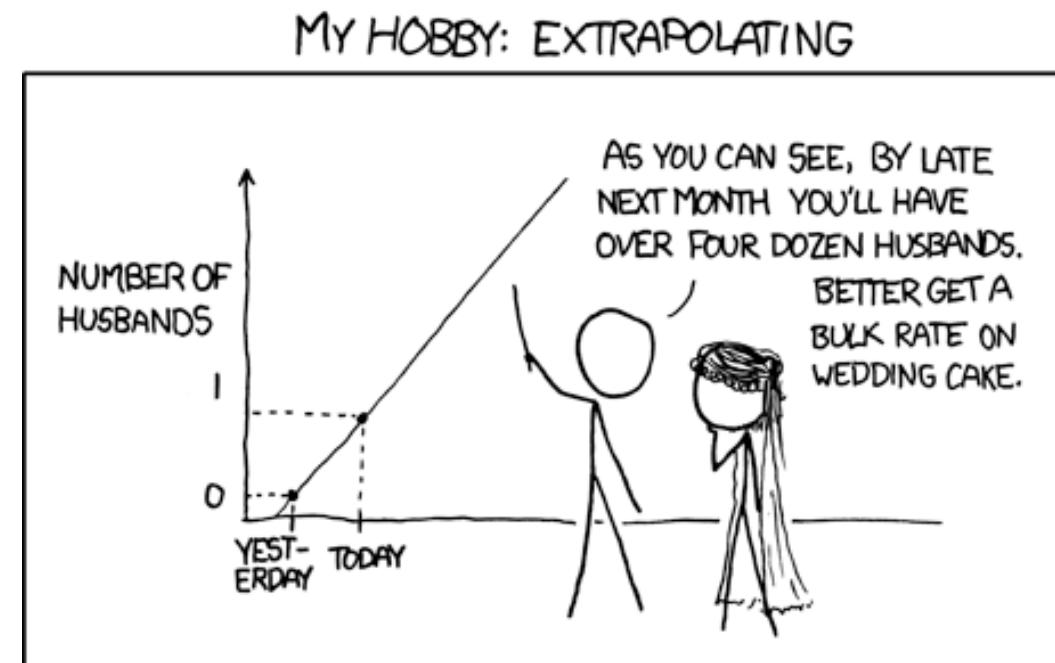
X

Foto nunca antes vista

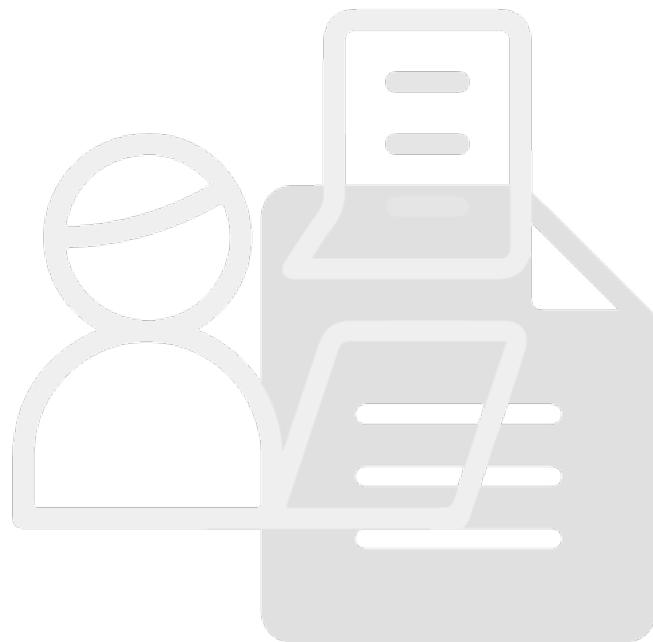


## Generalización

- El objetivo central del aprendizaje computacional es tener un buen desempeño en entradas nuevas, es decir, sobre entradas diferentes a las usadas cuando el modelo fue entrenado.
- La generalización se refiere a la capacidad de un modelo de tener un buen rendimiento en entradas no observadas previamente.



## Generalización

 Error de entrenamiento

- El conjunto de entrenamiento es el conjunto de datos con el cual se entrena un modelo de aprendizaje computacional.
- El error de entrenamiento se refiere al porcentaje de ejemplos de entrenamiento que el modelo de clasificación predice de manera incorrecta.
- Los algoritmos de entrenamiento buscan minimizar el error de entrenamiento.

## Generalización

 Error de generalización

- El error de generalización se define como el valor esperado del error de un modelo en entradas nuevas.
- El valor esperado se calcula sobre la distribución de entradas que esperamos que el sistema encuentre en la práctica.
- Es difícil calcular de manera exacta el error de generalización.
- Típicamente, el error de generalización se estima midiendo el error de un modelo entrenado sobre un conjunto de ejemplos de prueba.
- El conjunto de ejemplos de prueba debe ser diferente al conjunto de ejemplos de entrenamiento.

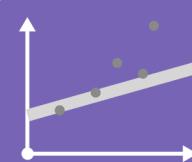
## Generalización

Sobreajuste, subajuste y ajuste apropiado

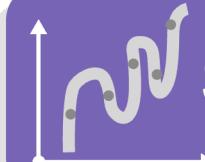
Un algoritmo de aprendizaje automático busca cumplir los siguientes objetivos:

- Hacer que el error de entrenamiento sea bajo.
- Hacer que la brecha entre el error de entrenamiento y el error de prueba sea pequeña.

Dependiendo del desempeño del algoritmo en estas dos habilidades se habla de que el modelo resultante tiene subajuste, sobreajuste o ajuste apropiado.



Subajuste



Sobreajuste



Ajuste apropiado

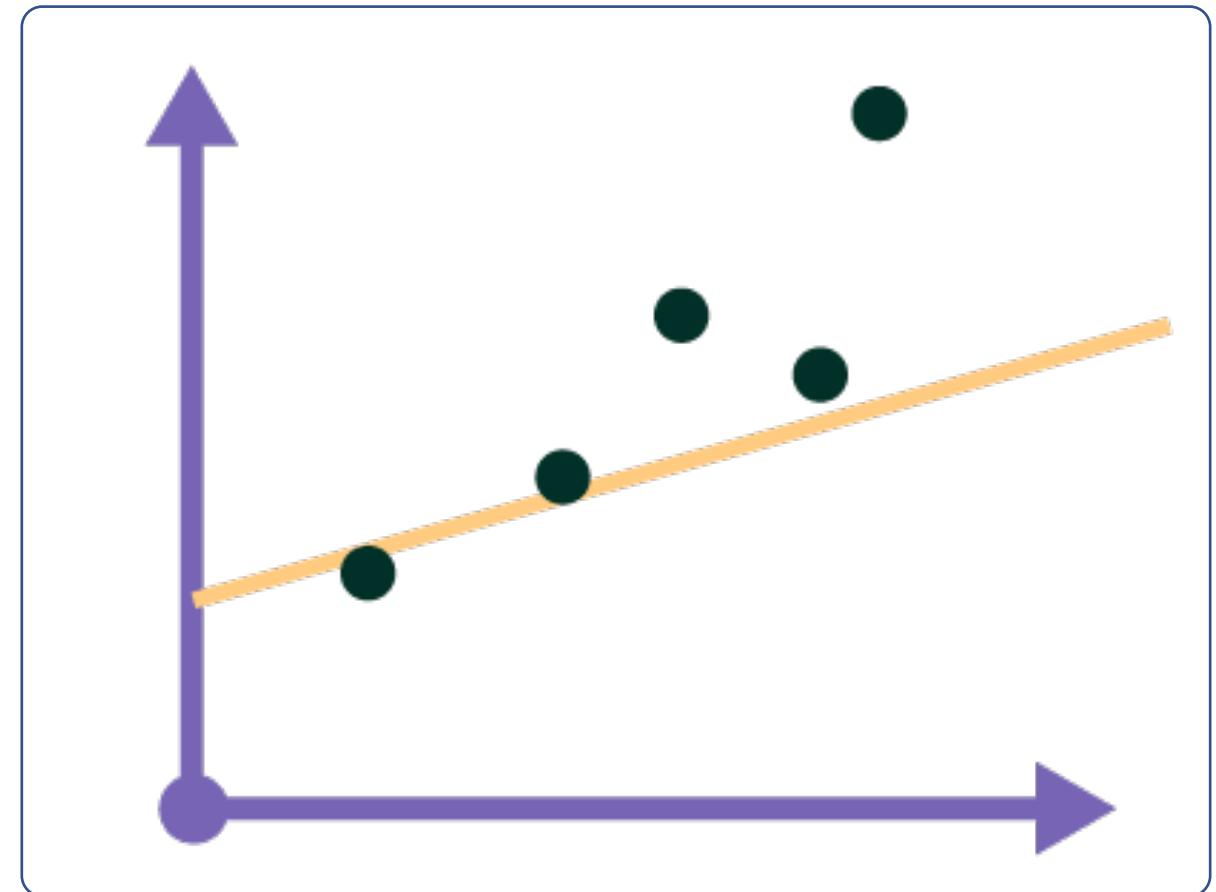
## Generalización

 Sobreajuste, subajuste y ajuste apropiado

## Subajuste

El **subajuste** ocurre cuando el modelo no logra conseguir un error de entrenamiento (ni de generalización) suficientemente bajo

Este resultado ocurre por que el modelo no tiene la capacidad suficiente para capturar la estructura de los datos y el problema.



## Generalización

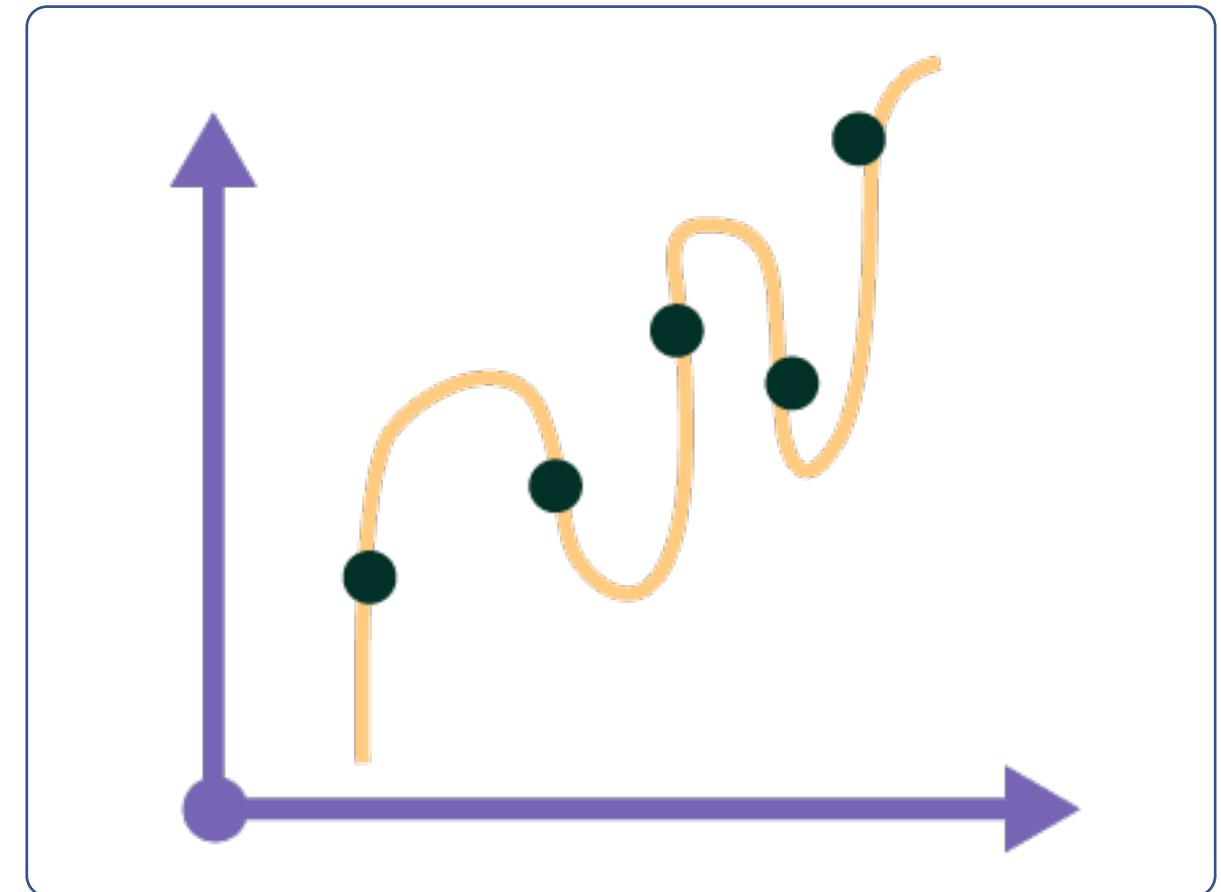
Sobreajuste, subajuste y ajuste apropiado

## Sobreajuste

El **sobreajuste** ocurre cuando la brecha entre el error de entrenamiento y el error de prueba es muy grande.

El error de entrenamiento es bajo pero el error de generalización es alto.

Esto resultado indica que el modelo se sobreajusta a las peculiaridades de los datos de entrenamiento.



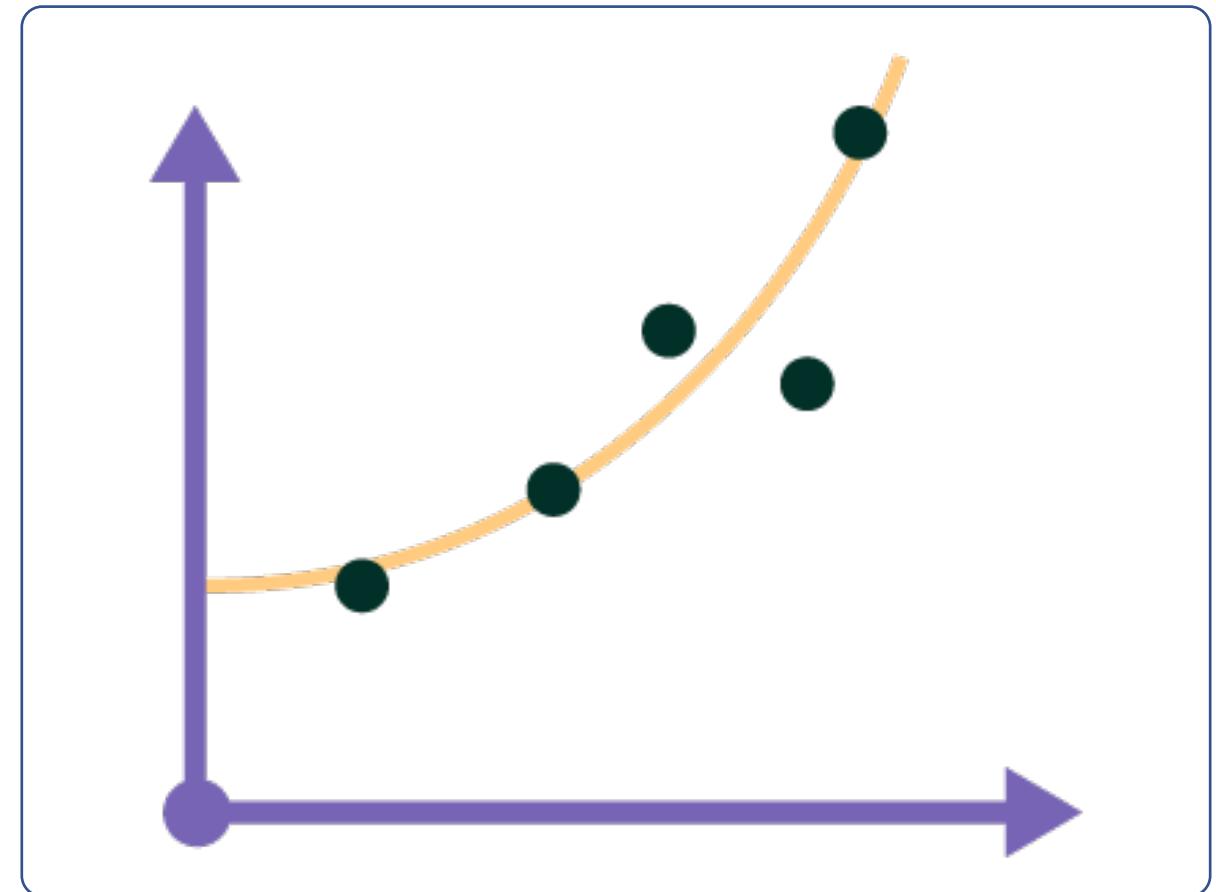
## Generalización

Sobreajuste, subajuste y ajuste apropiado

### Ajuste apropiado

El **ajuste apropiado** ocurre cuando tanto el error de entrenamiento y el de generalización son bajos.

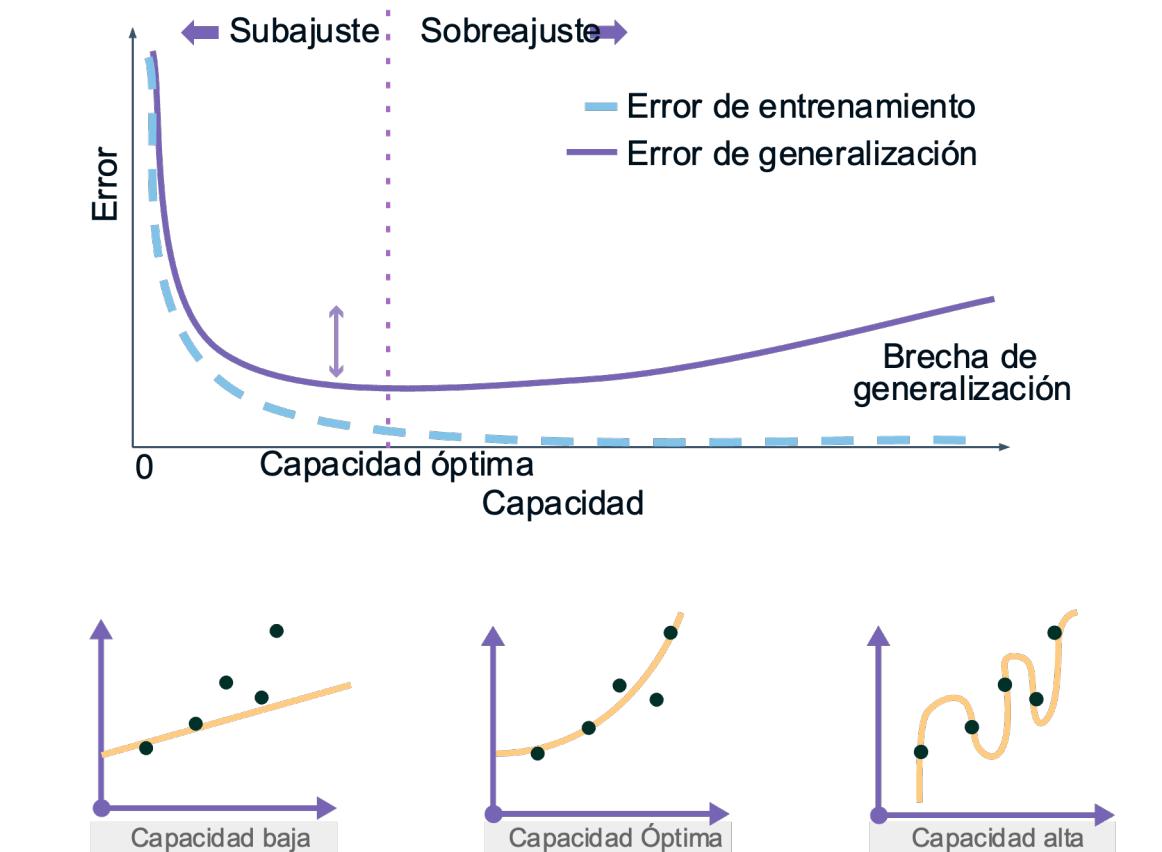
El modelo se desempeña de manera similar tanto en el conjunto de entrenamiento como en el conjunto de prueba y en ambos casos el desempeño es satisfactorio.



## Generalización

## Capacidad de un modelo de aprendizaje

- **Capacidad de un modelo:** habilidad de ajustarse a una amplia variedad de funciones.
- La capacidad de un algoritmo de aprendizaje se controla al elegir su **espacio de hipótesis**, el conjunto de funciones que el algoritmo de aprendizaje puede seleccionar como la solución.
- Por ejemplo, en un modelo de regresión polinomial, la capacidad se puede controlar especificando el máximo grado  $d$  del polinomio que se puede aprender.
- A mayor grado  $d$  más grande el espacio de hipótesis y por lo tanto mayor capacidad.

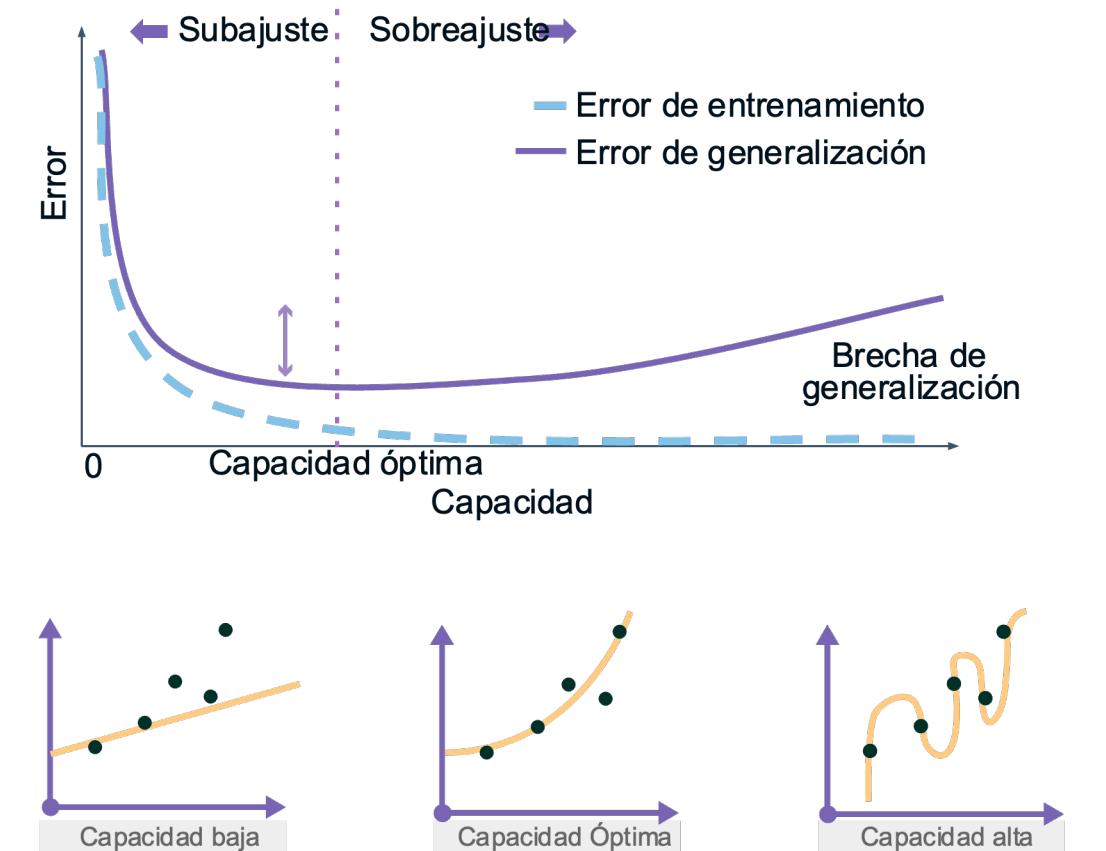


Fuente de la figura: adaptada de Goodfellow, .. y Pardo, s.f

## Generalización

## Capacidad de un modelo de aprendizaje

- En el lado izquierdo de la gráfica, ambos errores son altos, esta es la zona que corresponde al **subajuste**.
- A medida que aumentamos la capacidad, el error de entrenamiento se reduce, pero la brecha de generalización aumenta.
- Eventualmente, el tamaño de la brecha supera la disminución del error de entrenamiento, y se entra a la zona de **sobreajuste**, donde la capacidad es muy alta, por encima de la capacidad óptima.



Fuente de la figura: adaptada de Goodfellow, .. y Pardo, s.f

## Generalización



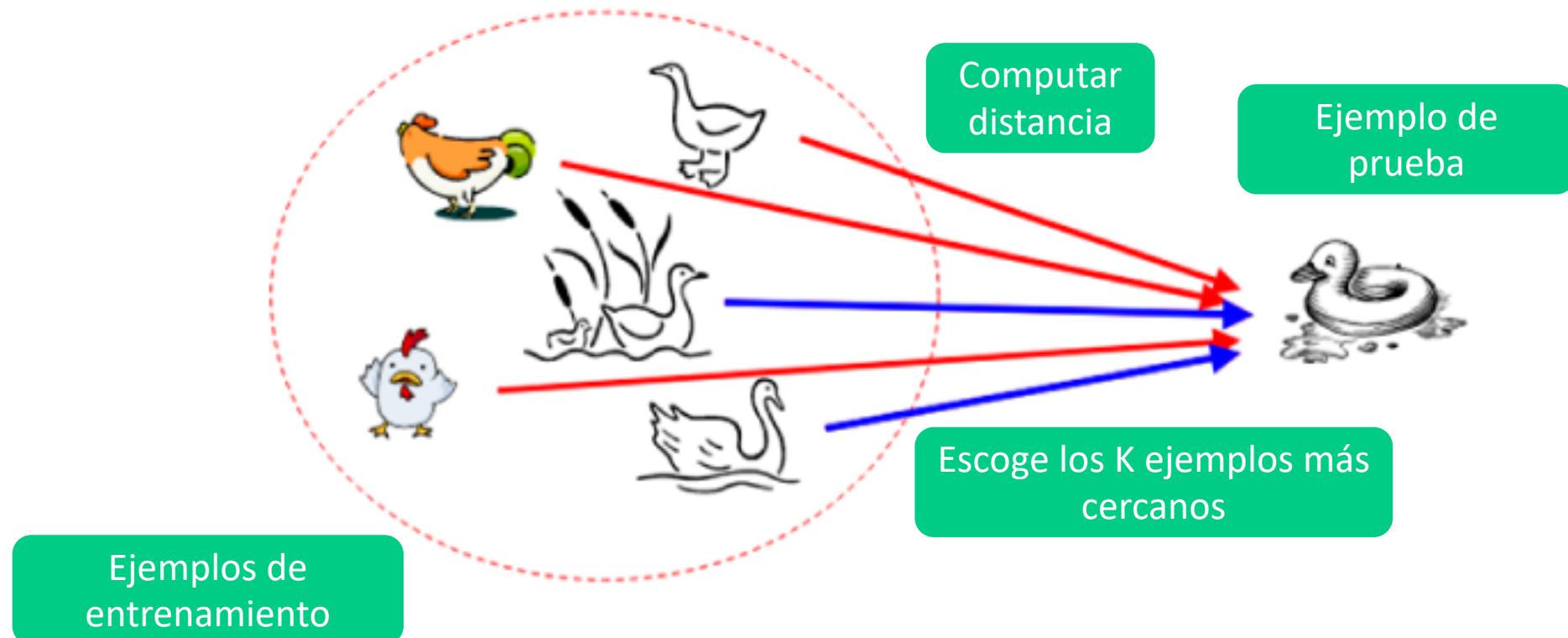
## Parámetros vs Hiperparámetros

- Los parámetros de un modelo se refieren a los valores que determinan la forma como el modelo funciona.
- Por ejemplo, en una regresión lineal los parámetros del modelo corresponden a los coeficientes por los cuales se multiplican las variables de entrada.
- Los parámetros son aprendidos por el algoritmo de aprendizaje a partir del conjunto de entrenamiento.
- Los hiperparámetros son las opciones que podemos usar para controlar el comportamiento del algoritmo de aprendizaje.
- Por ejemplo, el grado de un polinomio en regresión polinomial o la profundidad máxima de un árbol de decisión.
- Los hiperparámetros, en general, no se aprenden, sino que son especificados manualmente por el usuario.

## Clasificador K-NN

## Idea básica

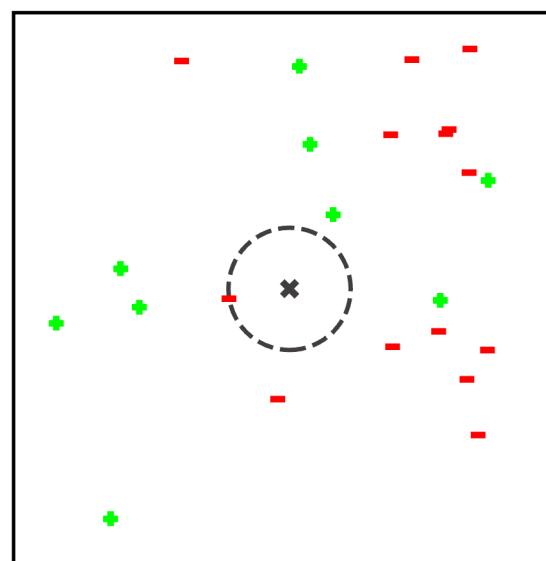
Si camina como un pato y suena como un pato, entonces, probablemente es un pato



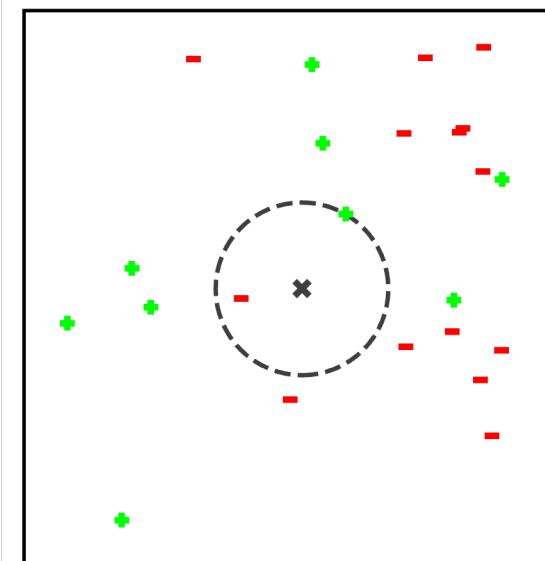
## Clasificador KNN

## Vecinos más cercano

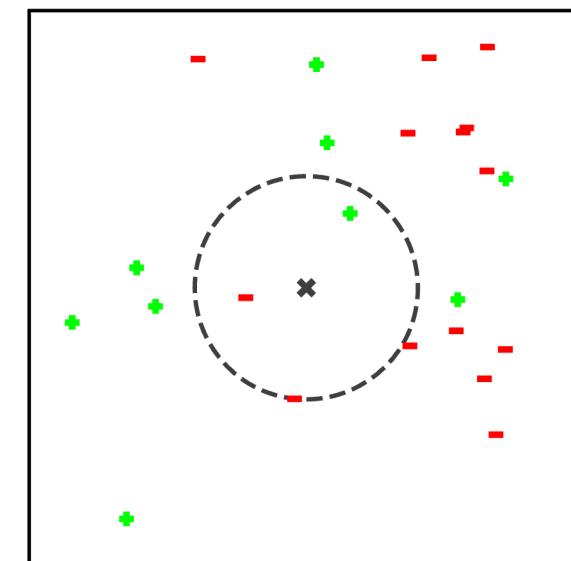
Los  $k$  ésimos vecinos más cercanos de un ejemplo  $x$  son los puntos de datos con la  $k$  éSIMA distancia más pequeña a  $x$ .



Primer vecino  
más cercano



Segundo vecino  
más cercano



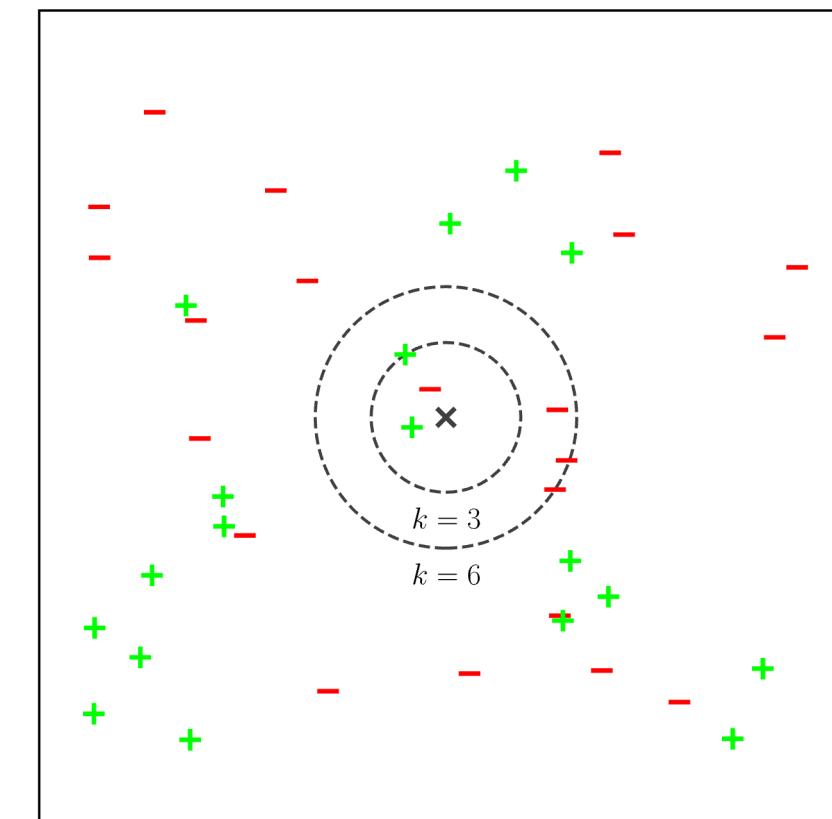
Tercer vecino  
más cercano

## Clasificador KNN

## Proceso de predicción

Para realizar el proceso de predicción se necesita lo siguiente:

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

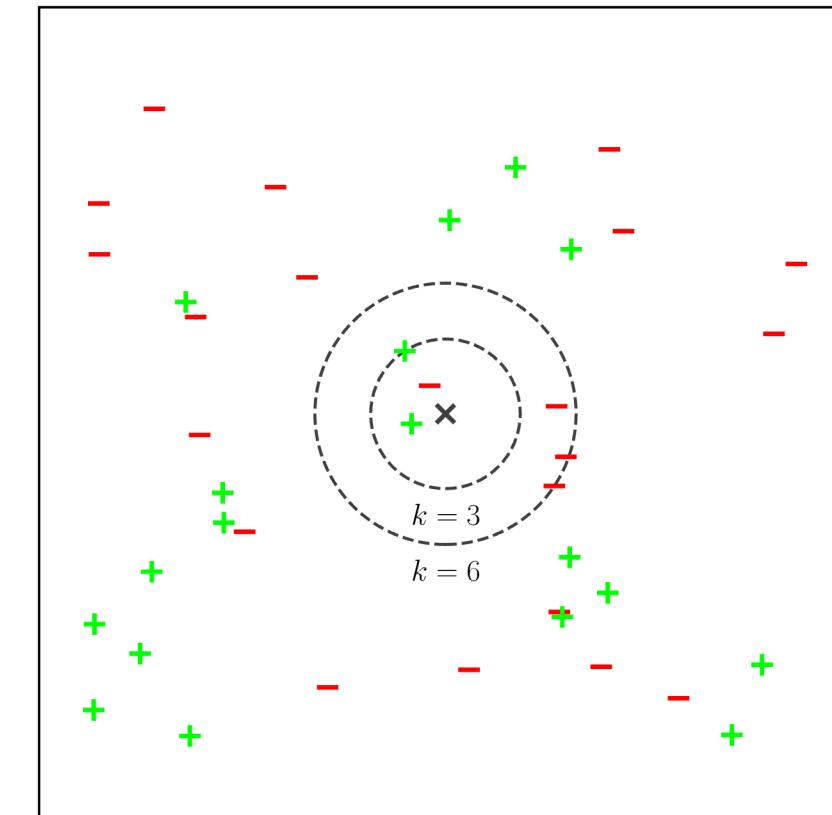


## Clasificador KNN

## 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.

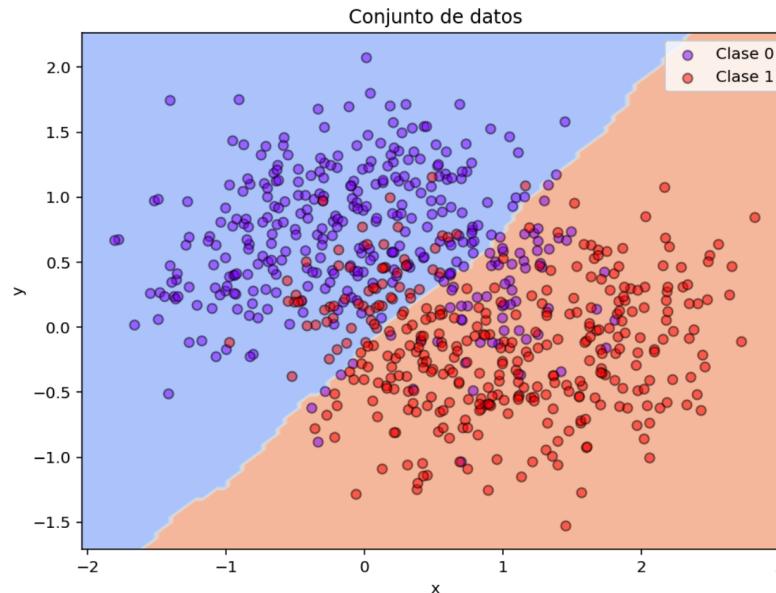
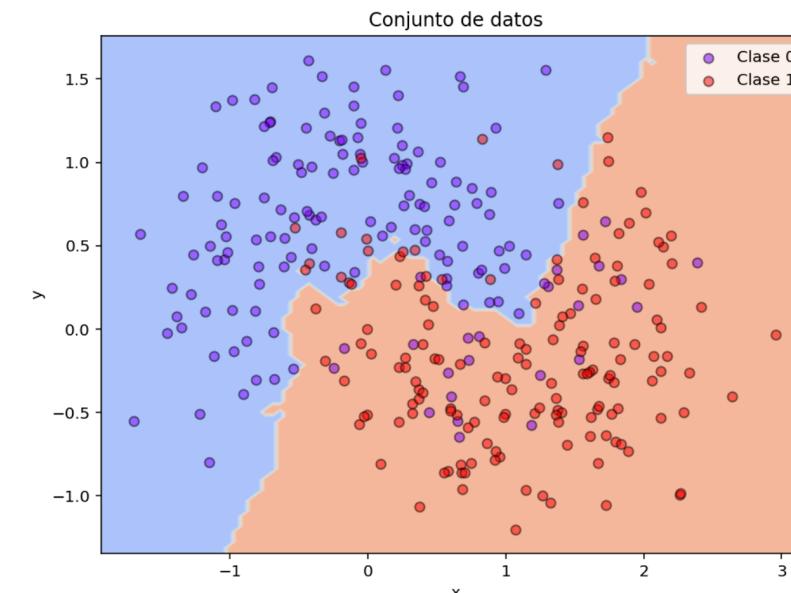
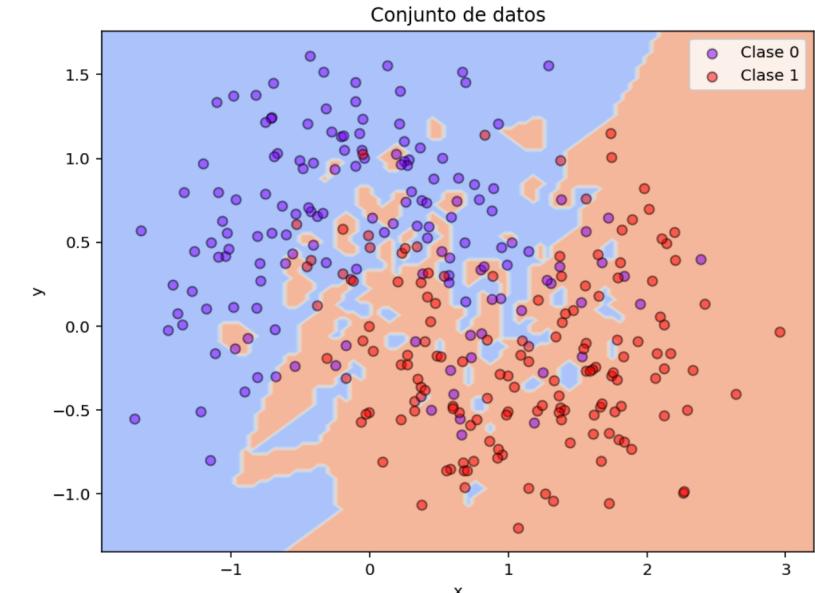


Ejemplo de clasificador KNN

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

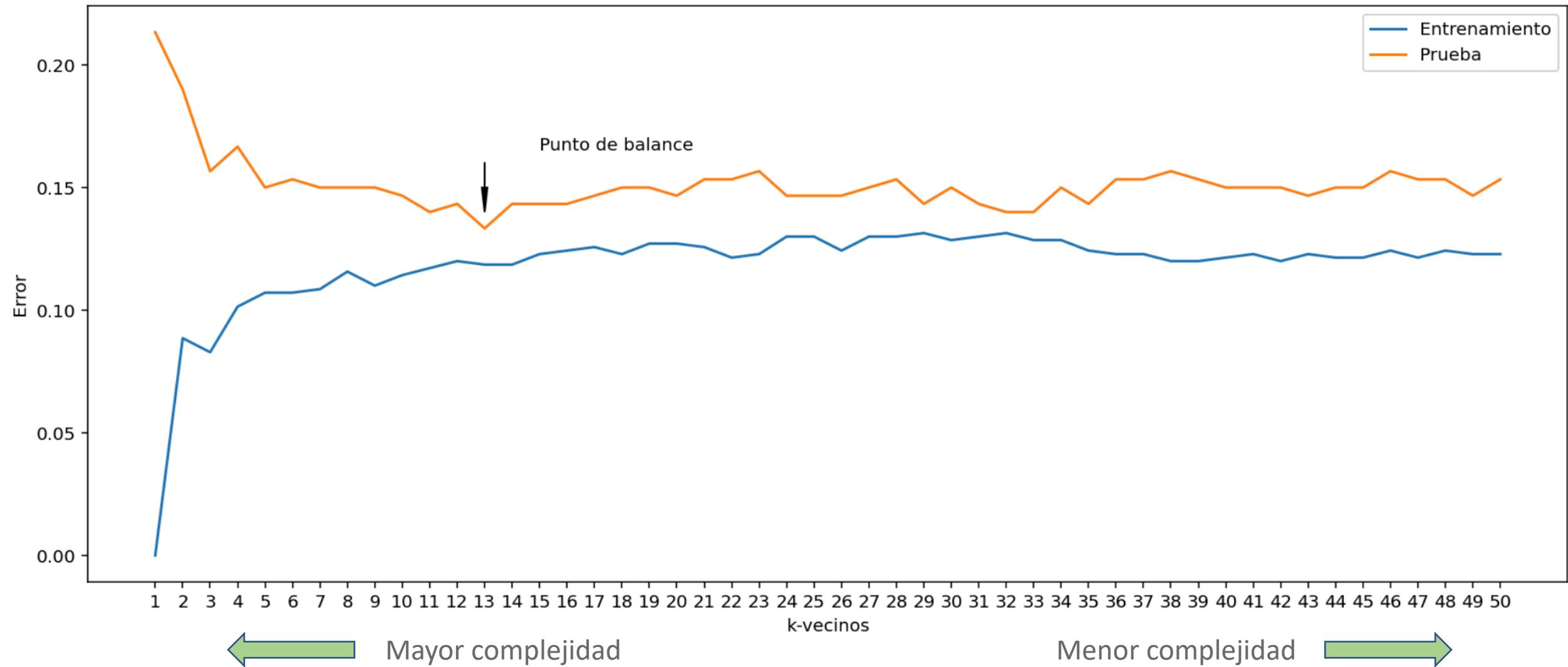
## Clasificador KNN

## Subajuste, ajuste apropiado y sobreajuste

 $k = 400$  $k = 13$  $k = 1$ 

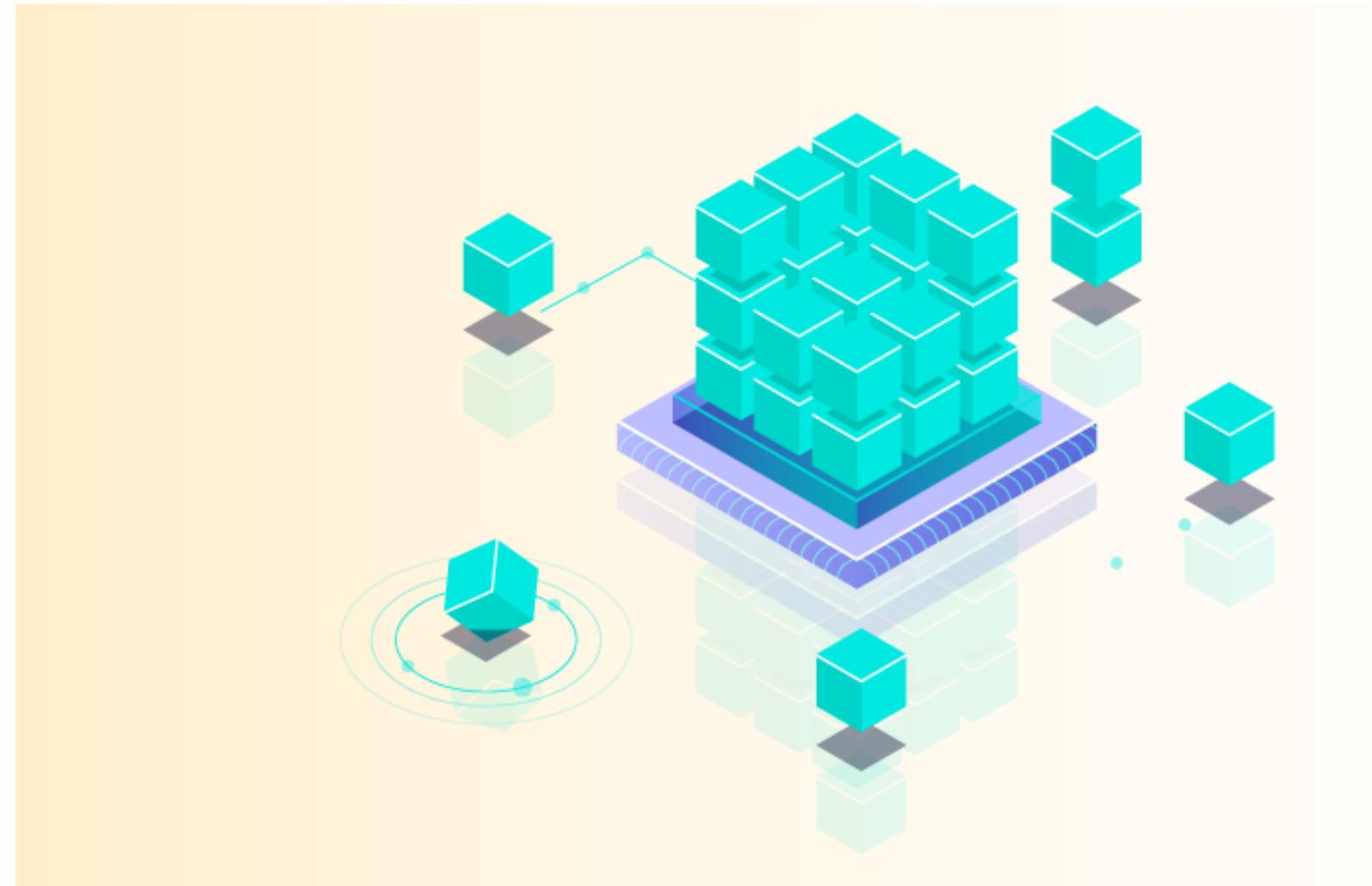
## Clasificador KNN

## Cálculo de la complejidad óptima



3

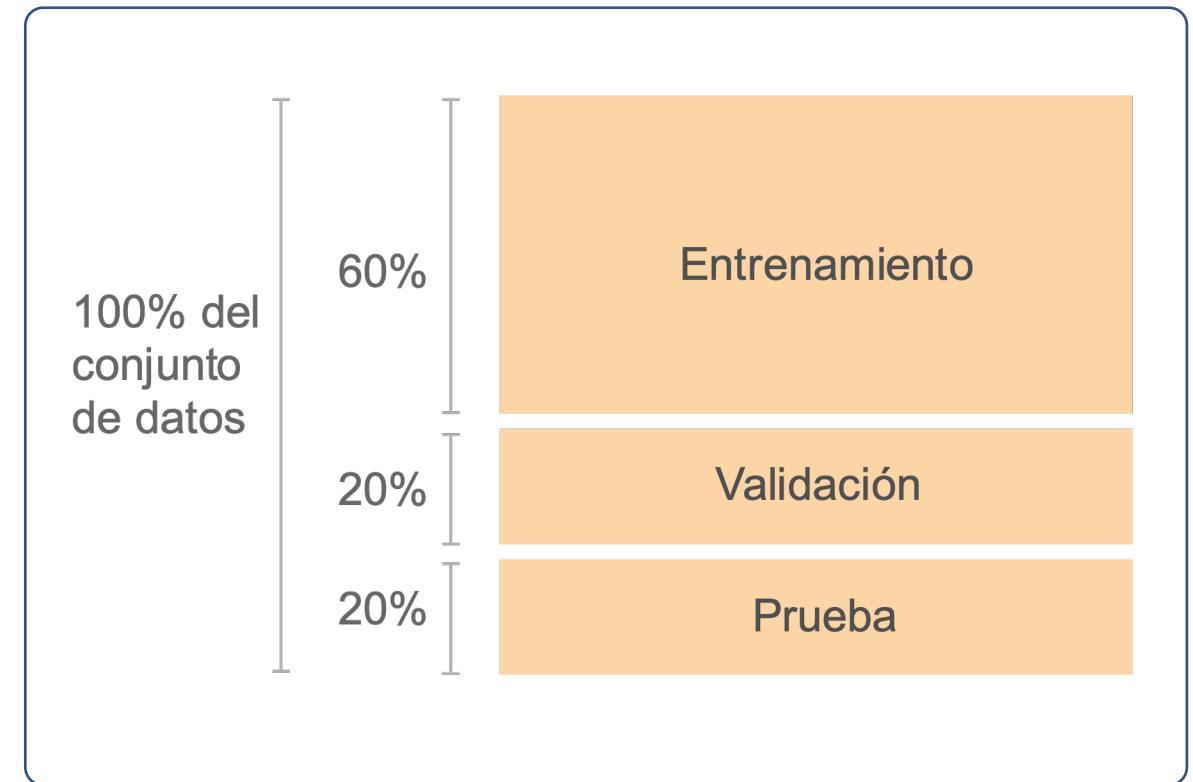
## Conjunto de entrenamiento, validación, prueba



## Conjunto de entrenamiento, validación, prueba

## Capacidad de un modelo de aprendizaje

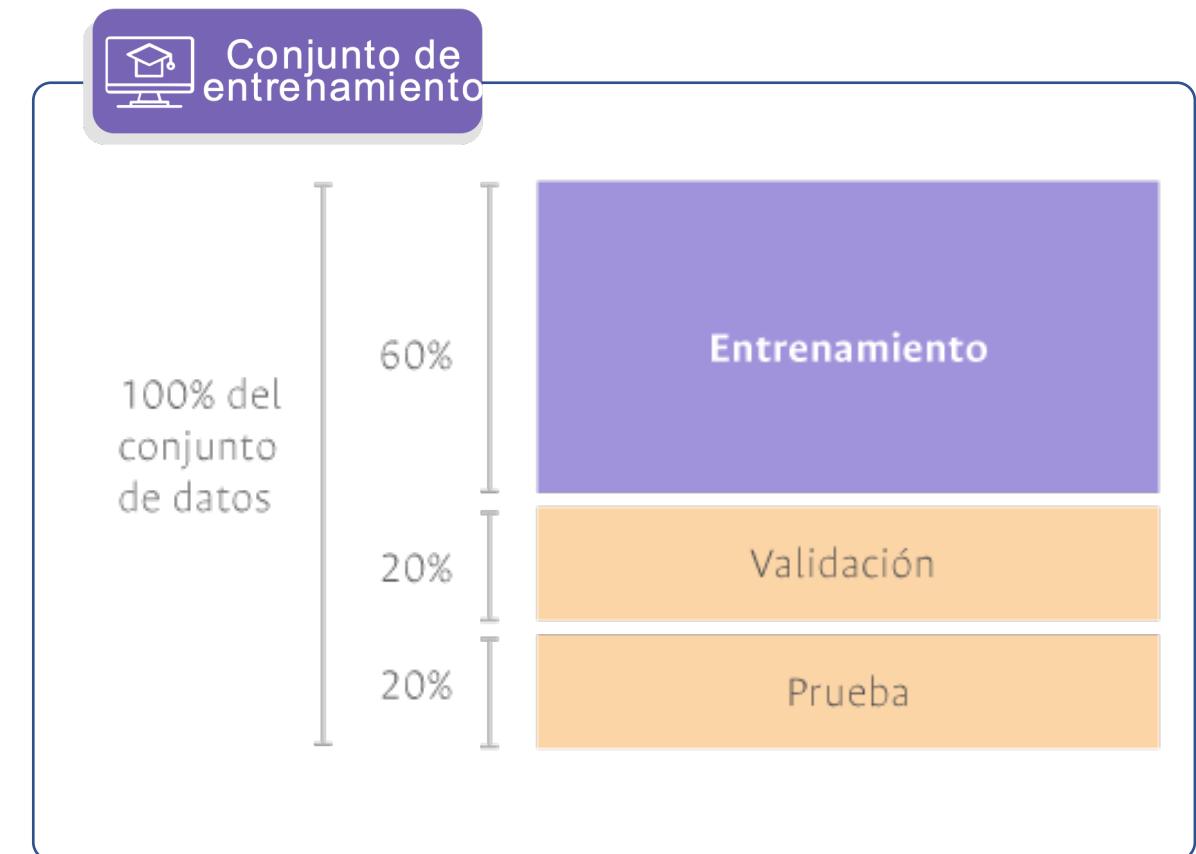
- En un experimento de aprendizaje computacional los datos disponibles se dividen generalmente en tres subconjuntos:
  - Entrenamiento
  - Validación
  - Prueba
- Una distribución típica es 60% para entrenamiento, 20% para validación y 20% para prueba.



Conjunto de entrenamiento, validación, prueba

## Conjunto de entrenamiento

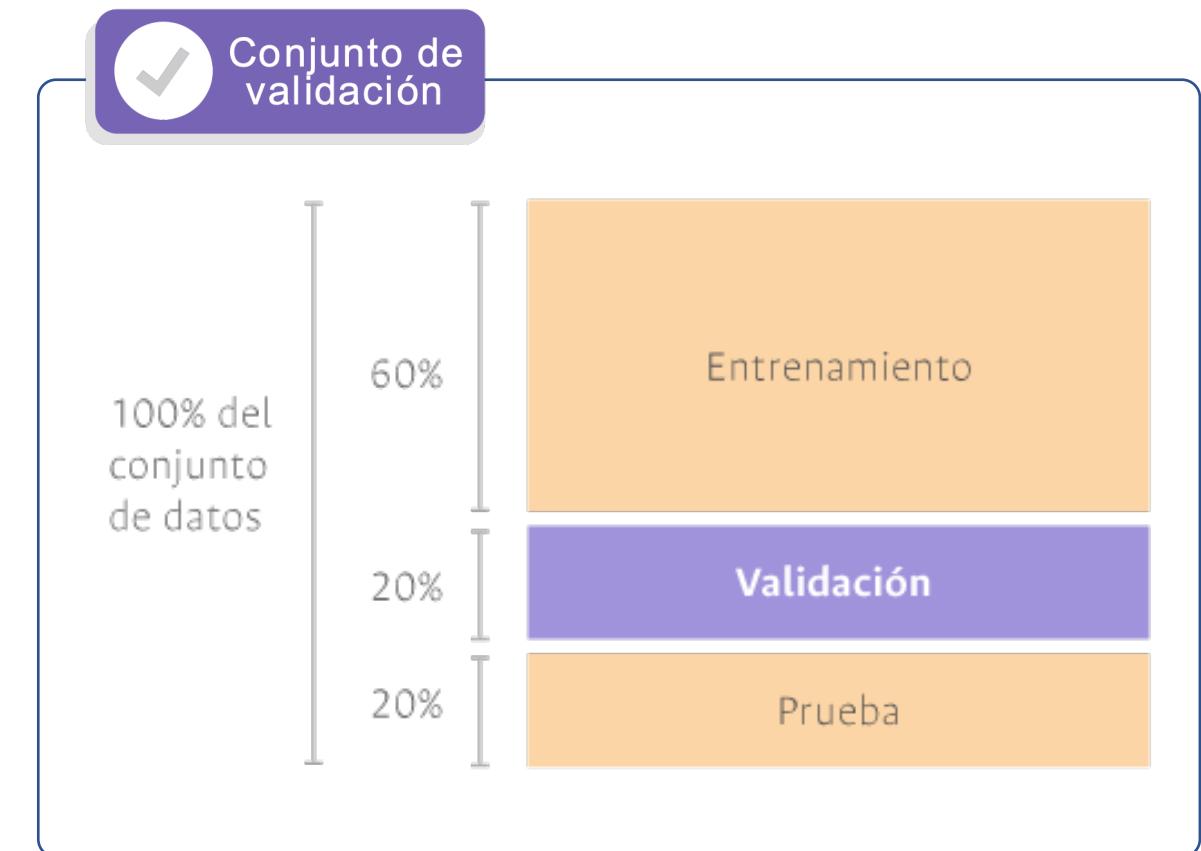
Es el subconjunto usado para entrenar el modelo, es decir, el subconjunto utilizado por el algoritmo de aprendizaje para aprender los parámetros del modelo.



## Conjunto de entrenamiento, validación, prueba

## Conjunto de validación

- Se utiliza para estimar el error de generalización.
- En cada iteración de la fase de entrenamiento, después de ajustar los parámetros, se calcula tanto el error de entrenamiento como el error de validación. A este proceso se le conoce como *Validación Cruzada*.
- La Validación Cruzada permite estimar el desempeño del modelo para diferentes configuraciones de los hiperparámetros.

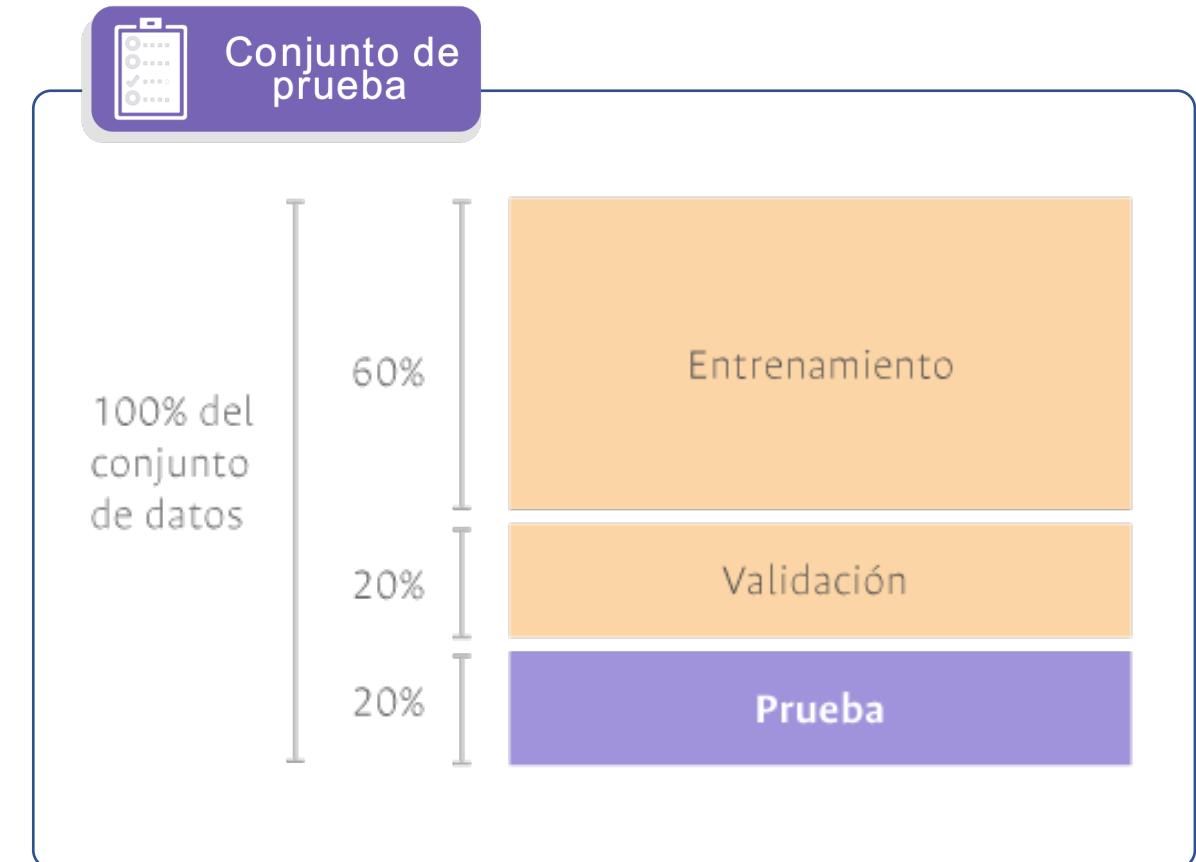


## Conjunto de entrenamiento, validación, prueba / Conjunto de prueba

Se utiliza para estimar el desempeño final del modelo entrenado; esta estimación permite prever cómo se comportará el modelo en el futuro con datos que nunca ha visto.

Asegúrese de que su conjunto de prueba reúna las siguientes dos condiciones:

- Ser lo suficientemente grande como para generar resultados significativos desde el punto de vista estadístico.
- Ser representativo de todo el conjunto de datos, en otras palabras, no elija un conjunto de prueba con características diferentes a las del conjunto de entrenamiento.



## Conjunto de entrenamiento, validación, prueba

## Uso de los subconjuntos de datos

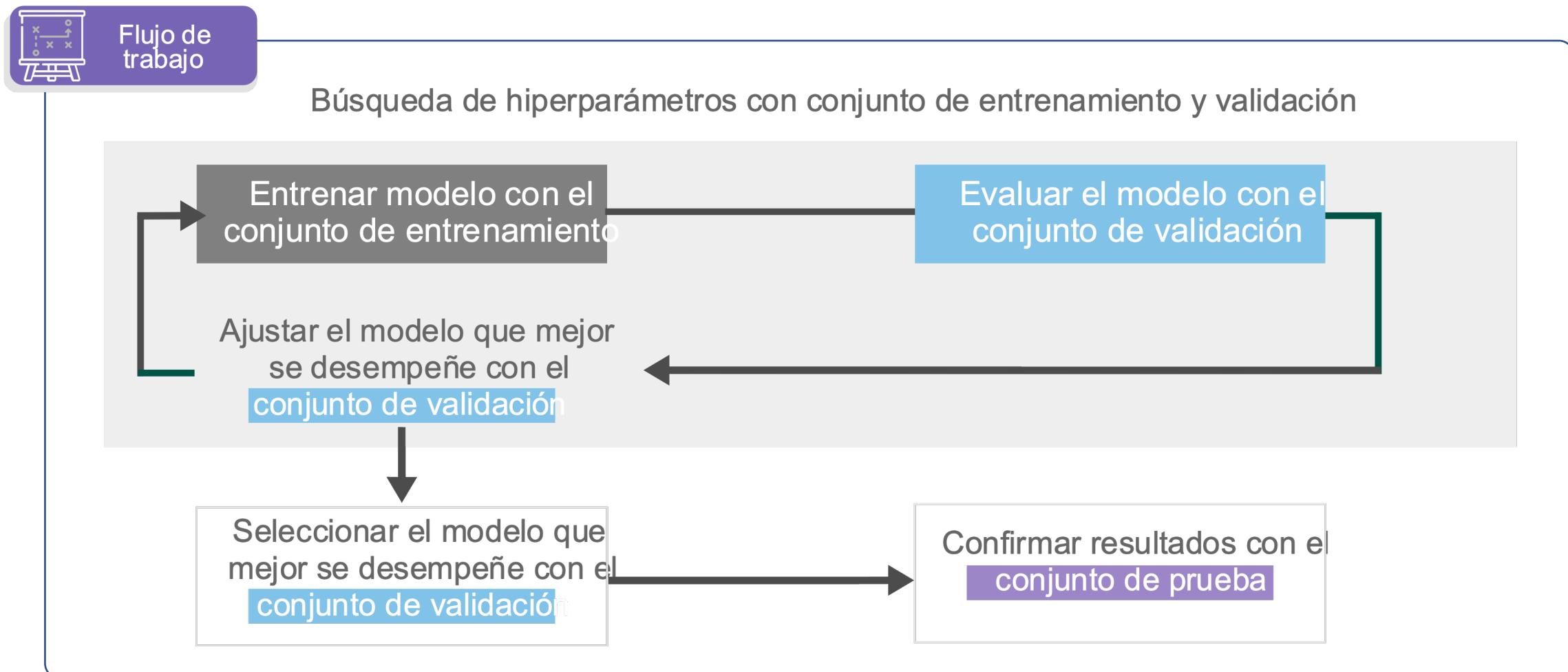


Figura. Flujo de trabajo con un conjunto de entrenamiento, prueba y validación

Fuente: adaptada de Google Developers (s.f.).

## Conjunto de entrenamiento, validación, prueba

 Validación cruzada de k-pliegues

- Al particionar los datos en 3 subconjuntos, reducimos drásticamente el número de ejemplos que pueden ser empleados para entrenar un modelo.
- Los resultados podrían depender de la selección aleatoria particular de la tupla (entrenamiento, validación).
- Una forma de mitigar este problema es realizar múltiples experimentos con diferentes conjuntos de entrenamiento y validación.
- La implementación más común de esta estrategia se llama *Validación Cruzada de k-pliegues*.



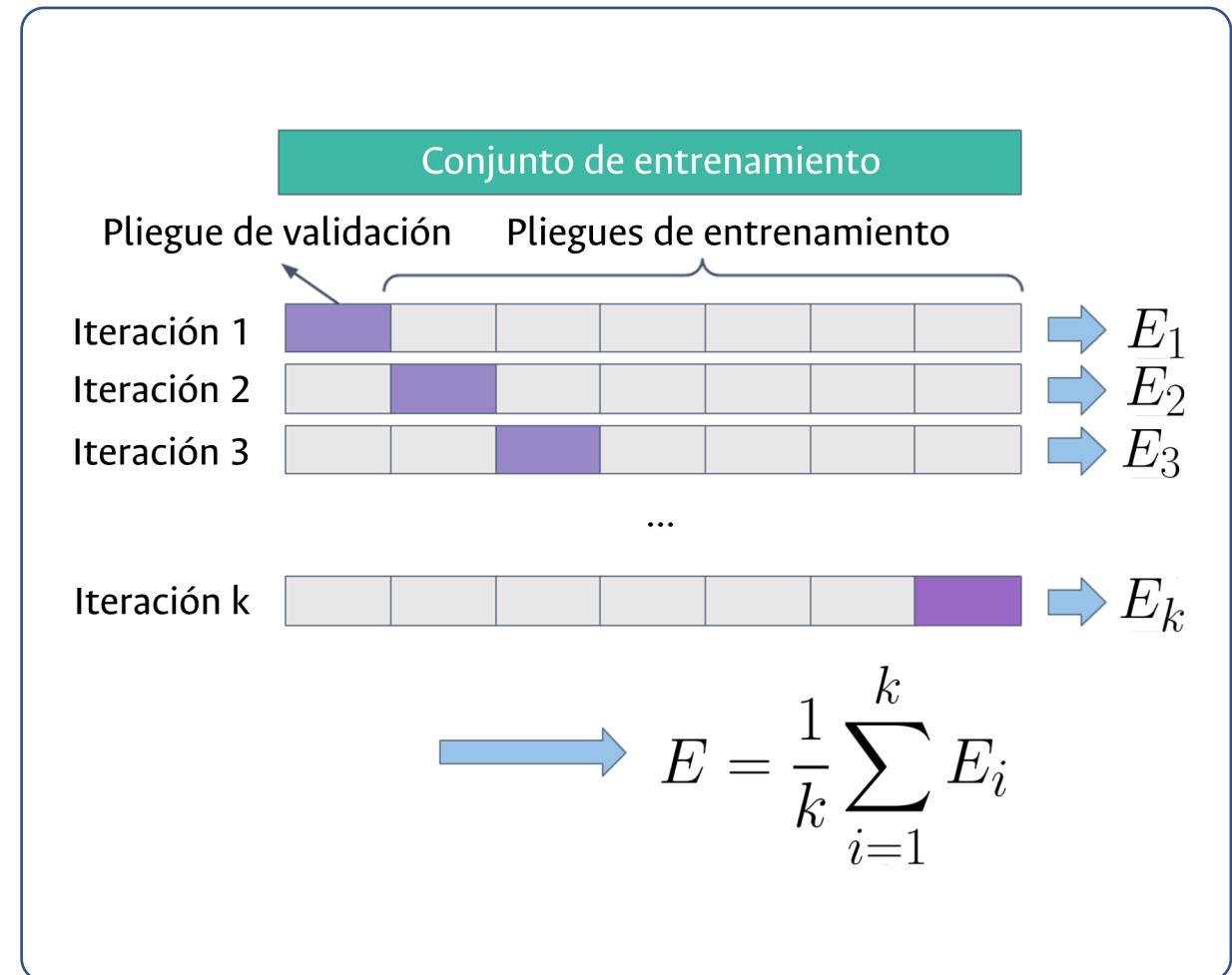
## Conjunto de entrenamiento, validación, prueba

## Validación cruzada de k-pliegues

La *Validación Cruzada de k-pliegues* partitiona el conjunto de entrenamiento en  $k$  conjuntos más pequeños y ejecuta el siguiente procedimiento por cada uno de los  $k$  pliegues:

- Un modelo es entrenado con los  $k - 1$  pliegues restantes
- El desempeño del modelo es evaluado en el pliegue que no fue usado para entrenar

Al final, el error de validación se obtiene como el promedio del error obtenido en cada iteración.



Conjunto de entrenamiento, validación, prueba

## Diseño de un experimento de aprendizaje computacional (Video)



## Métricas de desempeño



## Métricas de desempeño



Las métricas de desempeño permiten medir de manera objetiva que tan bien funciona el modelo sobre un conjunto de datos particular.

Para problemas de clasificación se suele usar la exactitud (accuracy):

$$\frac{\text{\# ejemplos correctamente clasificados}}{\text{\# total de ejemplos}}$$

Para problemas de regresión se suele usar el error cuadrático medio:

$$\frac{1}{n} \sum_{i=0}^n (x - \hat{x})^2$$

Donde:

x -Real  
 $\hat{x}$  -Predicho

## Métricas de desempeño

 Matriz de confusión

- La matriz de confusión muestra el desempeño de un método de clasificación sobre un conjunto de datos particular
- Permite visualizar cuantos datos de una determinada clase real son clasificados en las diferentes clases.
- Las filas de la matriz corresponden a la clase real de los datos
- Las columnas a la clase predicha por el modelo
- Los valores en cada posición corresponden al número de ejemplos clasificados

		Clase Predicha	
		Positivo	Negativo
Clase Real	Positivo	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (VN)

## Métricas de desempeño

## Métricas de desempeño para clasificación

Las métricas se definen a partir de las entradas dadas en la siguiente matriz de confusión.

		Clase Predicha	
		Positivo	Negativo
Clase Real	Positivo	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (VN)

Exactitud (*accuracy*)

$$\frac{VP + VN}{VP + FP + VN + FN}$$

## Métricas de desempeño

## Métricas de desempeño para clasificación

Las métricas se definen a partir de las entradas dadas en la siguiente matriz de confusión para un problema de 2 clases.

		Clase Predicha	
		Positivo	Negativo
Clase Real	Positivo	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (VN)

Error

$$1 - \frac{VP + VN}{VP + FP + VN + FN}$$

## Métricas de desempeño

## Métricas de desempeño para clasificación

Las métricas se definen a partir de las entradas dadas en la siguiente matriz de confusión.

		Clase Predicha	
		Positivo	Negativo
Clase Real	Positivo	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (VN)

Precisión

$$\frac{VP}{VP + FP}$$

## Métricas de desempeño

## Métricas de desempeño para clasificación

Las métricas se definen a partir de las entradas dadas en la siguiente matriz de confusión.

		Clase Predicha	
		Positivo	Negativo
Clase Real	Positivo	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (VN)

Especificidad

$$\frac{VN}{VN + FP}$$

## Métricas de desempeño

## Métricas de desempeño para clasificación

Las métricas se definen a partir de las entradas dadas en la siguiente matriz de confusión.

		Clase Predicha	
		Positivo	Negativo
Clase Real	Positivo	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (VN)

Sensibilidad(*recall*)

$$\frac{VP}{VP + FN}$$

## Métricas de desempeño

 Métricas de desempeño para clasificación

Las métricas se definen a partir de las entradas dadas en la siguiente matriz de confusión.

		Clase Predicha	
		Positivo	Negativo
Clase Real	Positivo	Verdaderos positivos (VP)	Falsos negativos (FN)
	Negativo	Falsos positivos (FP)	Verdaderos negativos (VN)

 Valor-F (*F1-score*)

$$2 \cdot \frac{\text{precisión} \cdot \text{sensibilidad}}{\text{precisión} + \text{sensibilidad}}$$

## Métricas de desempeño

## Métricas de desempeño para regresión

## Error Cuadrático Medio (MSE)

$$\frac{1}{n} \sum_{i=0}^n (\hat{y}_i - y_i)^2$$

## Raíz del Error Cuadrático Medio (RMSE)

$$\sqrt{\frac{1}{n} \sum_{i=0}^n (\hat{y}_i - y_i)^2}$$

## R2 - Coeficiente de Determinación

$$R^2 = 1 - \frac{\sum_{i=0}^n (\hat{y}_i - y_i)^2}{\sum_{i=0}^n (y_i - \bar{y})^2}$$



## Despedida

# Fabio Augusto Gonzalez, PhD.

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

[fagonzalezo@unal.edu.co](mailto: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

Scikit-learn:

[Scikit-learn: Machine Learning in Python](#), Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.



## Recursos adicionales

### Aprendizaje Computacional

Alpaydin, E. (2010). Introduction to Machine Learning. [Introducción al aprendizaje de máquinas]  
[https://kkpatel7.files.wordpress.com/2015/04/alppaydin\\_machinelearning\\_2010.pdf](https://kkpatel7.files.wordpress.com/2015/04/alppaydin_machinelearning_2010.pdf)

Fabio. A. Gonzalez. (2020). Machine Learning [Curso Aprendizaje Computacional] <https://fagonzalezo.github.io/ml-2020-1/>

Fabio. A. Gonzalez. (2020). Introducción a los Sistemas Inteligentes <https://fagonzalezo.github.io/iis-2020-1/>

### Bibliografía adicional

Mayo, M. (Mayo de 2018). Marcos para abordar el proceso de aprendizaje automático. Kdnuggets.  
<https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html>

Mayo, M. ( s.f.). El proceso de ciencia de datos, redescubierto. Kdnuggets. <https://www.kdnuggets.com/2016/03/data-science-process-rediscovered.html>

Google developers. (s.f.). Introducción a la estructura de problemas de aprendizaje automático.  
<https://developers.google.com/machine-learning/problem-framing>



## Derechos de imágenes

Scikit-learn. (s.f). Imágenes sección scikit-learn. [Logo]. <https://scikit-learn.org/stable/>

Scikit-learn. (s.f). Logo scikit-learn. [Logo]. <https://scikit-learn.org/stable/>

Pixabay. (s.f). Aprendizaje computacional cerebro. [Vector]. <https://pixabay.com/vectors/machine-learning-information-brain-5433370/>

Freepik. (s.f). ordenador portátil pantalla blanca teclado. [Fotografía]. [https://www.freepik.es/vector-gratis/ordenador-portatil-pantalla-blanca-teclado\\_7222477.htm#page=1&query=computador&position=1](https://www.freepik.es/vector-gratis/ordenador-portatil-pantalla-blanca-teclado_7222477.htm#page=1&query=computador&position=1)

Pixabay. (s.f). Estudiante adolescente con libro. [Vector]. <https://pixabay.com/vectors/student-teenager-book-learning-147783/>

Pixabay. (s.f). Dedo pulgar positivo. [Vector]. <https://pixabay.com/vectors/thumb-positive-finger-excellent-1429327/>

Pixabay. (s.f). Dedo pulgar negativo. [Vector]. <https://pixabay.com/vectors/thumb-bad-down-minus-failure-1429333/>

Pixabay. (s.f). Red neuronal. [Vector]. <https://pixabay.com/vectors/neural-network-thought-mind-mental-3816319/>



## 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

