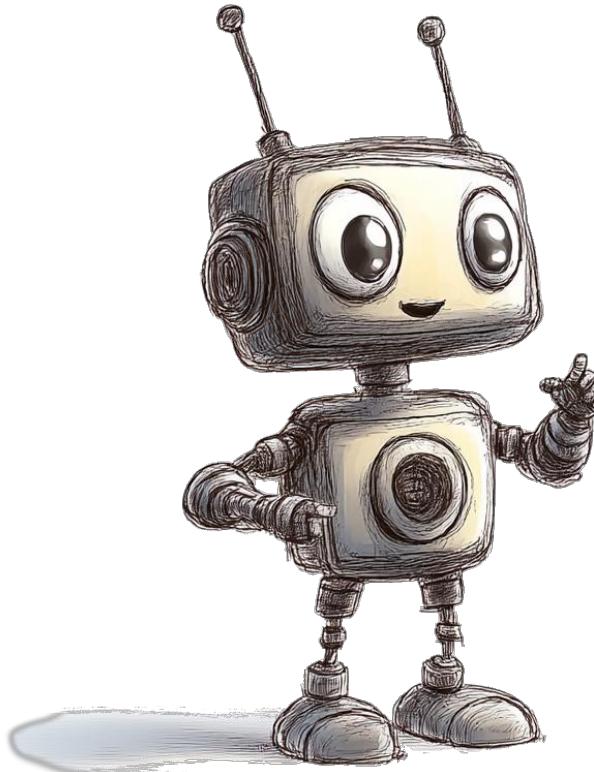


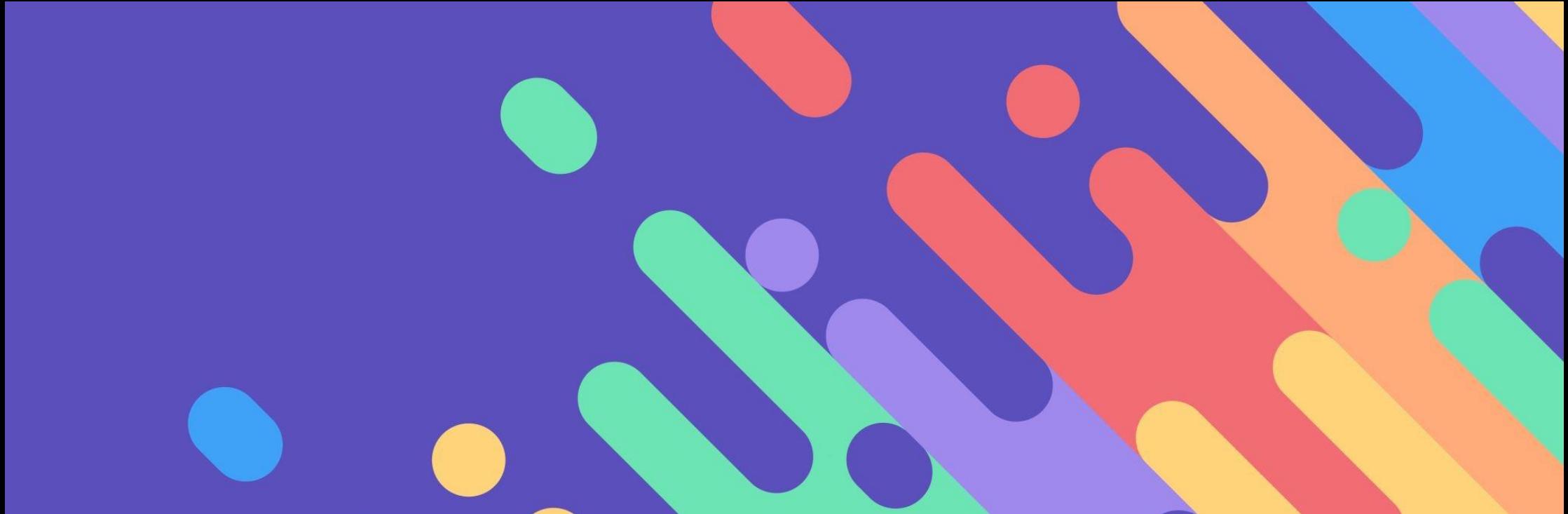
---

# APRENDIZAJE AUTOMÁTICO



Los temas que vamos a ver en este video son:

- Definición de aprendizaje automático
- Datos
  - Tipos de datos
- Formas que las maquinas aprenden
  - Aprendizaje supervisado
  - Aprendizaje no supervisado
  - Aprendizaje por refuerzo



---

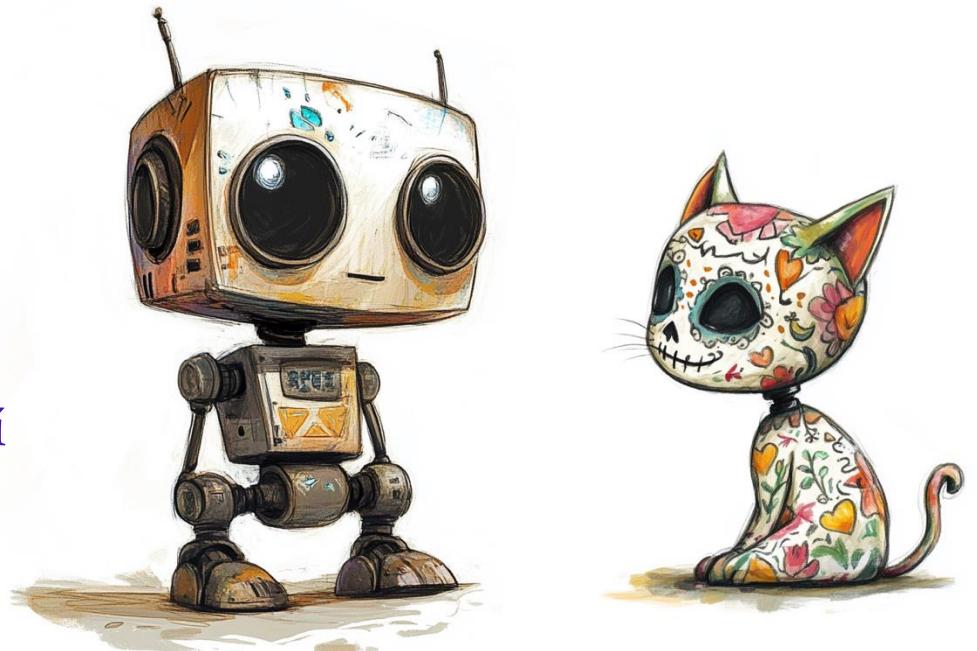
# APRENDIZAJE AUTOMÁTICO

# APRENDIZAJE AUTOMÁTICO

*Una computadora observa algunos datos, construye un modelo basado en estos datos y usa el modelo como una hipótesis sobre el mundo y como una pieza de software que puede resolver problemas.*

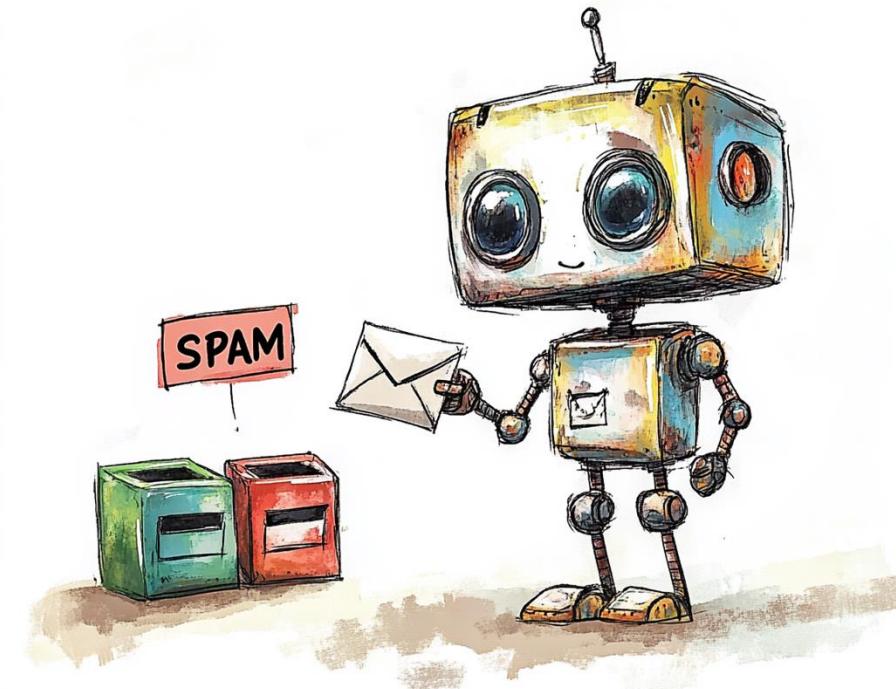
¿Pero por qué queremos que una computadora aprenda?

- No se pueden anticipar todas las posibles situaciones futuras.
- No se tiene idea de cómo programar una solución por sí mismo.



# APRENDIZAJE AUTOMÁTICO – EJEMPLOS

## Detección de SPAM

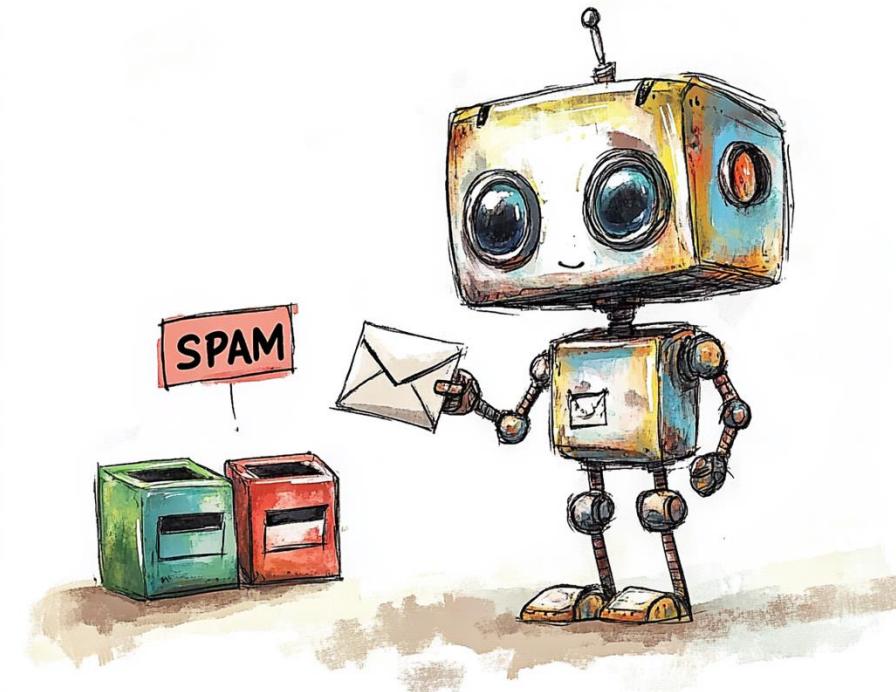


Este problema se resuelve usando Aprendizaje Automático. Podemos ir analizando las palabras y signos de puntuación de los correos.

Este es un problema que llamamos de *aprendizaje supervisado*, cuyo resultado es un indicador si el correo es SPAM o no. Esto se llama *problema de clasificación*.

# APRENDIZAJE AUTOMÁTICO – EJEMPLOS

## Detección de SPAM



Este problema se resuelve usando Aprendizaje Automático. Podemos ir analizando las palabras y signos de puntuación de los correos.

Este es un problema que llamamos de *aprendizaje supervisado*, cuyo resultado es un indicador si el correo es SPAM o no. Esto se llama *problema de clasificación*.

En este problema no todo error al predecir es lo mismo. *Queremos evitar filtrar el buen correo electrónico*, mientras que dejar pasar el spam no es deseable pero sus consecuencias son menos graves.

# APRENDIZAJE AUTOMÁTICO – EJEMPLOS

Precio de venta de una casa

Se intenta predecir el precio de venta más apropiado de una casa dadas diferentes características, tales como tamaño, edad, y lugar donde se encuentra.

Este es un problema de *aprendizaje supervisado*, conocido como *problema de regresión*, porque la medición del resultado es *cuantitativa*.



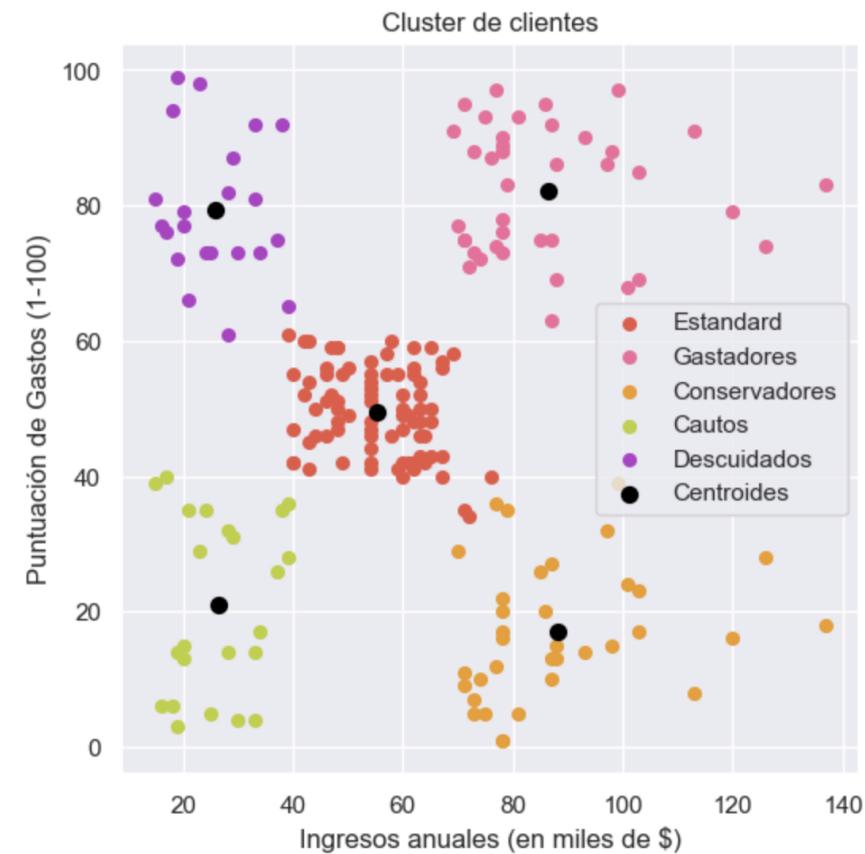
# APRENDIZAJE AUTOMÁTICO – EJEMPLOS

## Descubrimiento de segmentos de clientes.

Una famosa cadena de electrodomésticos quiere entender cómo se segmentan sus clientes en información demográfica y gustos, dados por su historial de compra.

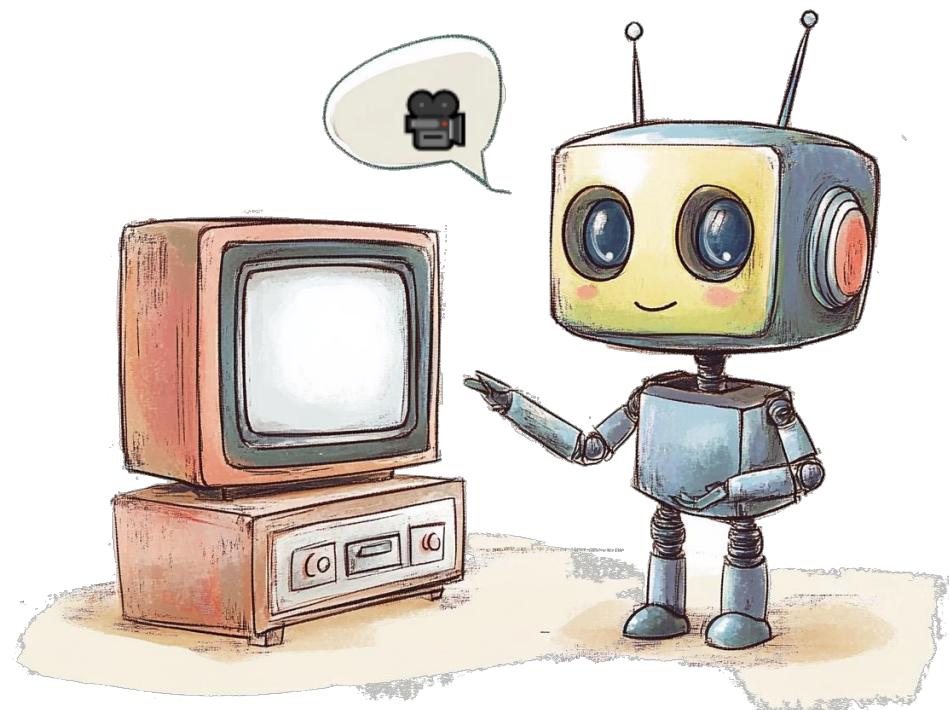
Usando estos datos, se emplea un modelo que agrupa los datos en diferentes grupos por *similitud*.

Este tipo de problema es de *aprendizaje no supervisado* y particularmente de *agrupamiento (clustering)*.



# APRENDIZAJE AUTOMÁTICO – EJEMPLOS

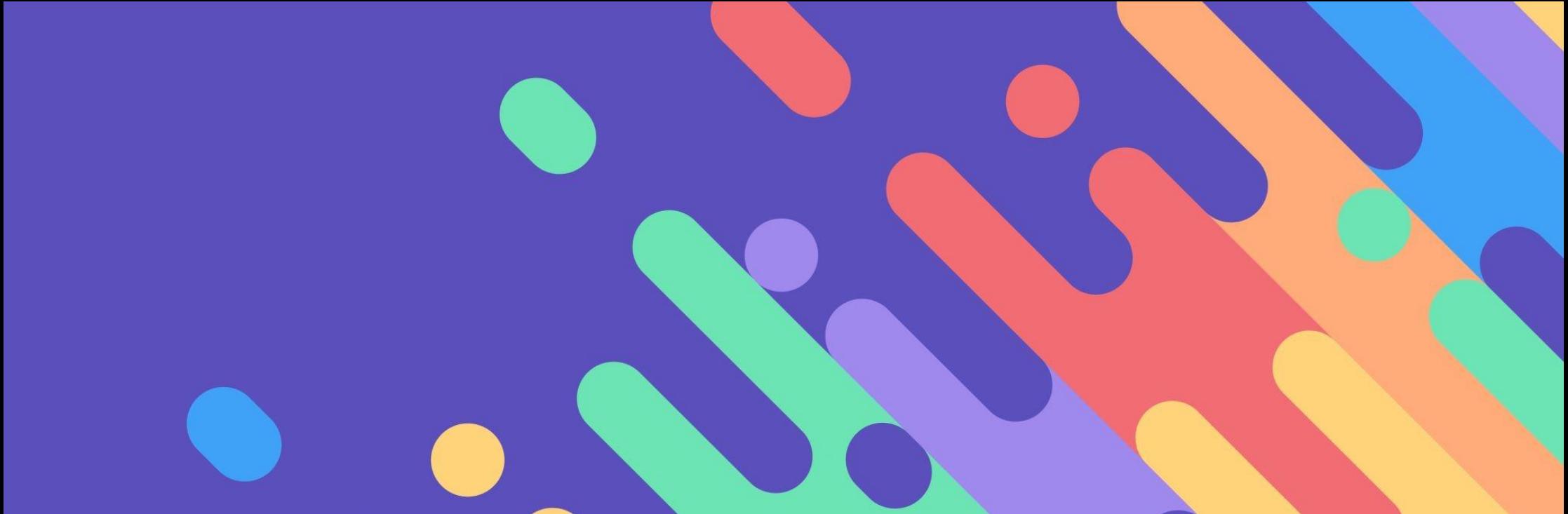
## Sistema de recomendación de videos



Una empresa dedicada a compartir videos necesita mantener a los usuarios comprometidos en la plataforma.

Necesita que, cuando un nuevo video termine, se le recomiende al usuario el siguiente video a ver, de tal forma que la persona siga deseando mantenerse en la plataforma.

Este tipo de algoritmos se llaman *sistemas de recomendación* y pueden englobarse en *aprendizaje supervisado, aprendizaje no supervisado o en aprendizaje por refuerzo*.



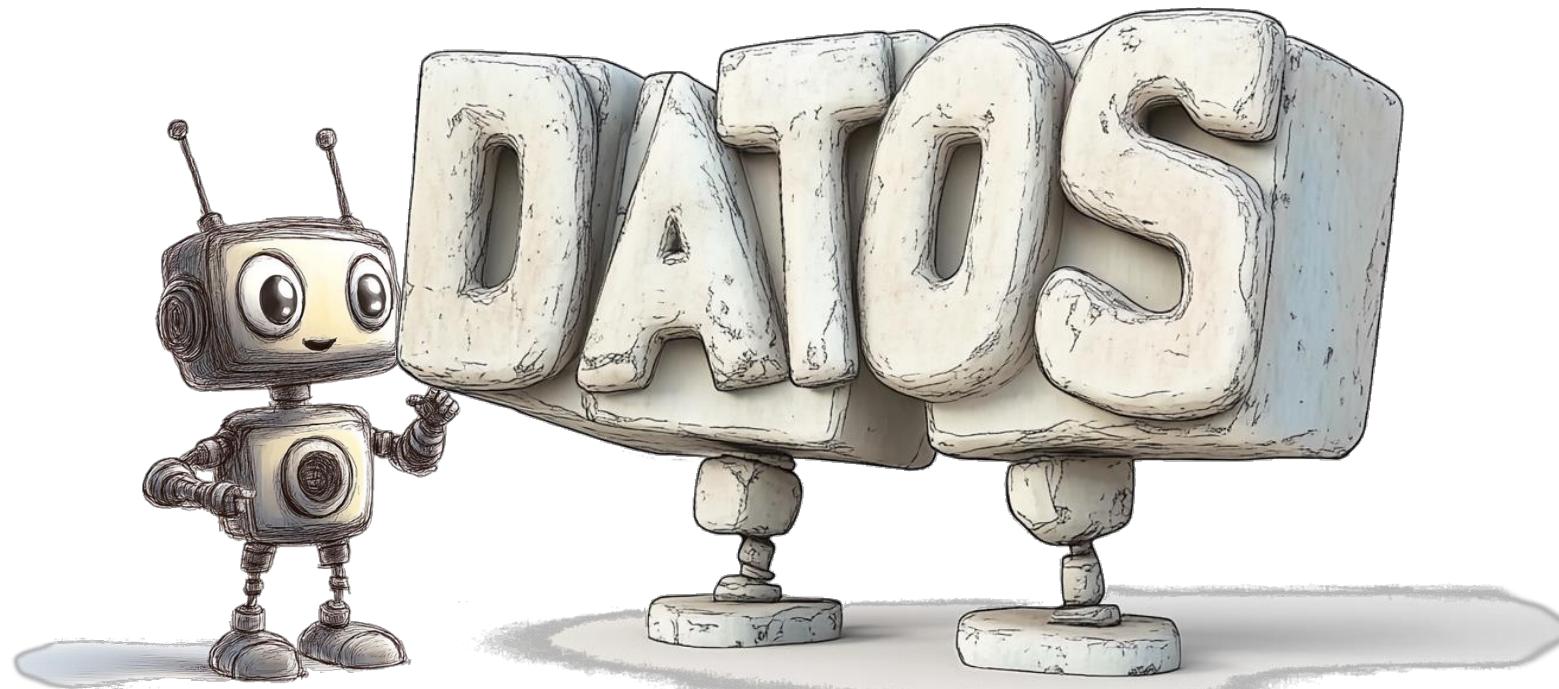
---

# DATOS

---

# DATOS

Lo más importante en Aprendizaje Automático son los...



# DATOS

Nos permite describir un objeto al que podemos llamar **entidad**.

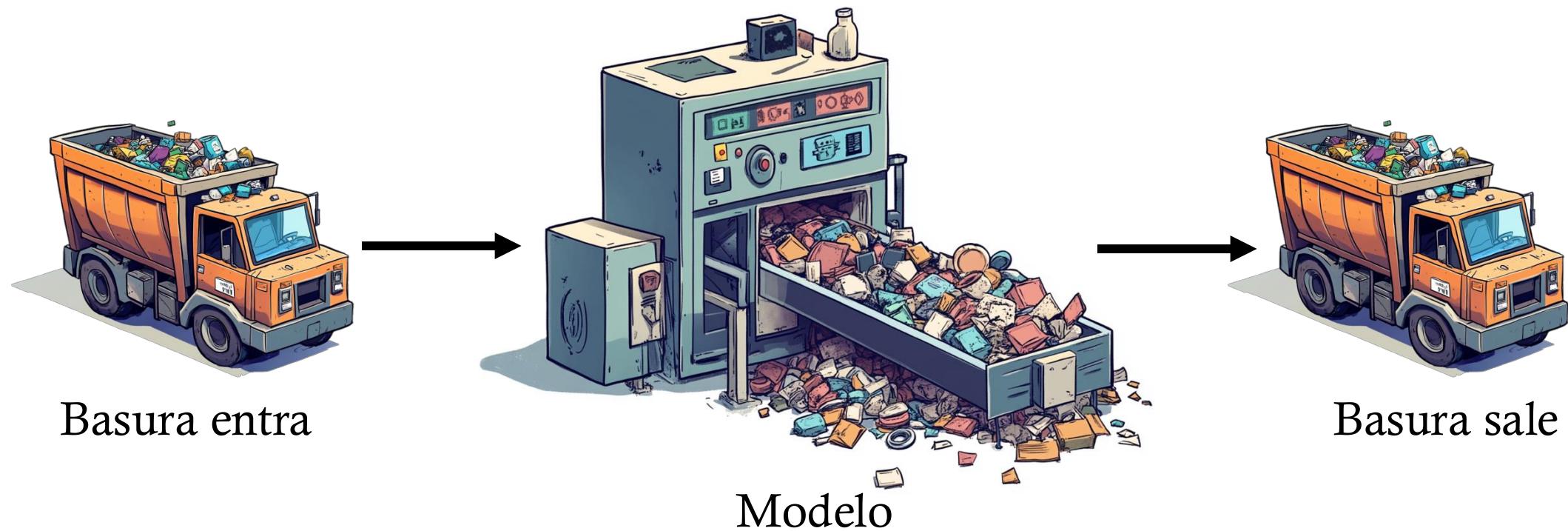
Esta entidad y su información pueden ser diferentes a pesar de que describan un mismo objeto.

*La forma en que se elija representar los datos no solo afecta la forma en que se construyen sus sistemas, sino también los problemas que sus sistemas pueden resolver.*

Por ejemplo, queremos representar un auto:



# DATOS



---

# DATOS

Los datos se pueden encontrar en dos formas:

- **Datos estructurados:** Tienen un formato estandarizado que permite tanto al software como a las personas acceder a estos de forma eficaz. Se trata **de datos tabulares con filas y columnas que definen claramente sus atributos**. Las computadoras pueden procesar eficazmente los datos estructurados en busca de información, dado que se trata de información cuantitativa.
- **Datos no estructurados:** Son información sin un **modelo de datos** establecido o datos que no están ordenados de una manera predefinida. Por ejemplo, archivos de texto, video, informes, e-mails, imágenes.

---

# DATOS

## Datos estructurados

En aprendizaje automático en general, se usan estructuras de dos dimensiones y notaciones vectoriales para referirnos a los datos:

- Cada fila del array es una **muestra**, **observación** o dato puntual.
- Cada columna es una **característica** (**feature** o **atributo**) de la observación.
- En el caso más general, hay una columna que llamaremos **objetivo**, **label**, **etiqueta** o **respuesta**, y que será el valor que se pretende predecir.

# DATOS

Atributos/features					Objetivo
Position	Experience	Skill	Country	City	Salary (\$)
Developer	0	1	Argentina	Buenos Aires	103100
Data Scientist	2	2	Uruguay	Montevideo	104900
Developer	3	1	Argentina	Chivilcoy	106800
QA Eng	2	2	Colombia	Bogotá	108700
Product Manager	1	5	Perú	Lima	110400
Developer	7	5	Paraguay	Asunción	112300
Cloud Eng	5	2	Argentina	Buenos Aires	116100

# DATOS

Cuando modelamos usando aprendizaje automático tenemos un variable dependiente (**target**) **y** (podemos conocer o no), dado un conjunto de predictores **X<sub>1</sub>, X<sub>2</sub>, ..., X<sub>p</sub>**, el cual podemos describir como un vector **X = (x<sub>1</sub>, x<sub>2</sub>, ..., x<sub>p</sub>)**.

*El desafío en aprendizaje automático es elegir las entradas correctas.*

El conjunto total de estas observaciones se llama **población**, lo cual casi nunca podemos tener; sino que tenemos un subconjunto de estas observaciones.

Sub-set

Población

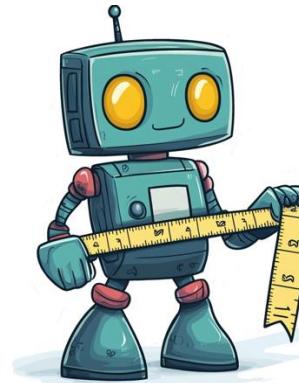
# DATOS

Tenemos diferentes tipos de variables:

**Variables numéricas:** Son aquellas que representan números y con ellas se pueden realizar operaciones aritméticas.



Discretas



Continuas

# DATOS

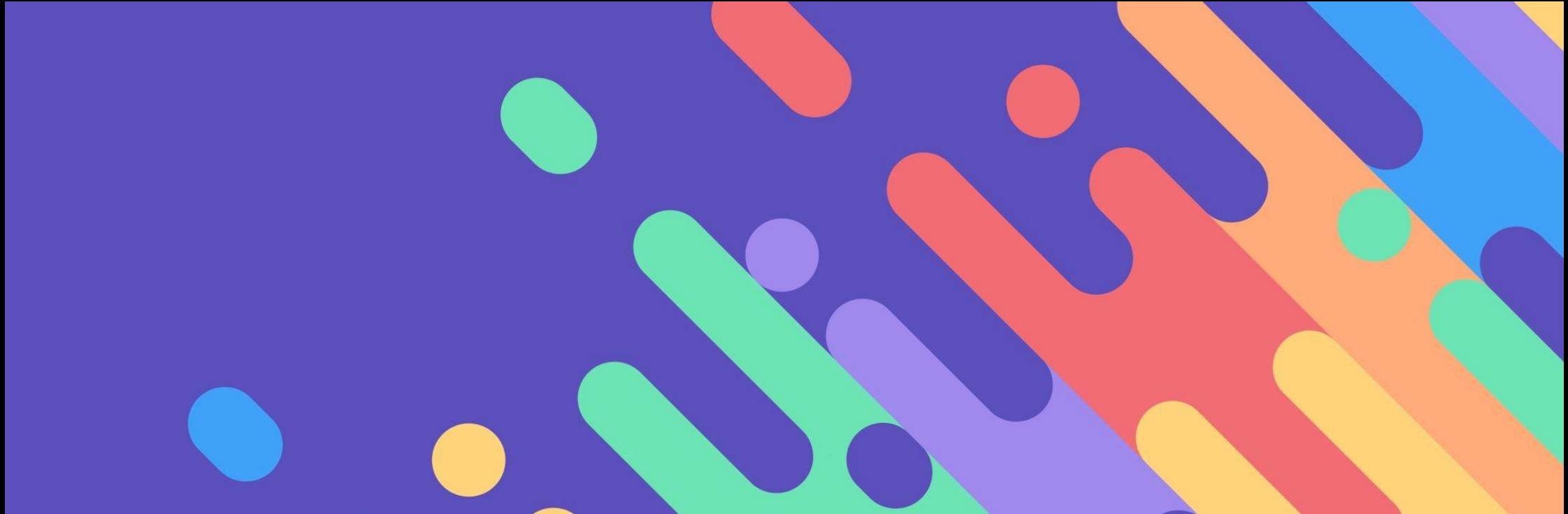
**Variables categóricas:** Es una variable que puede tomar uno de un número limitado y, por lo general, fijo de posibles valores.



Nominal



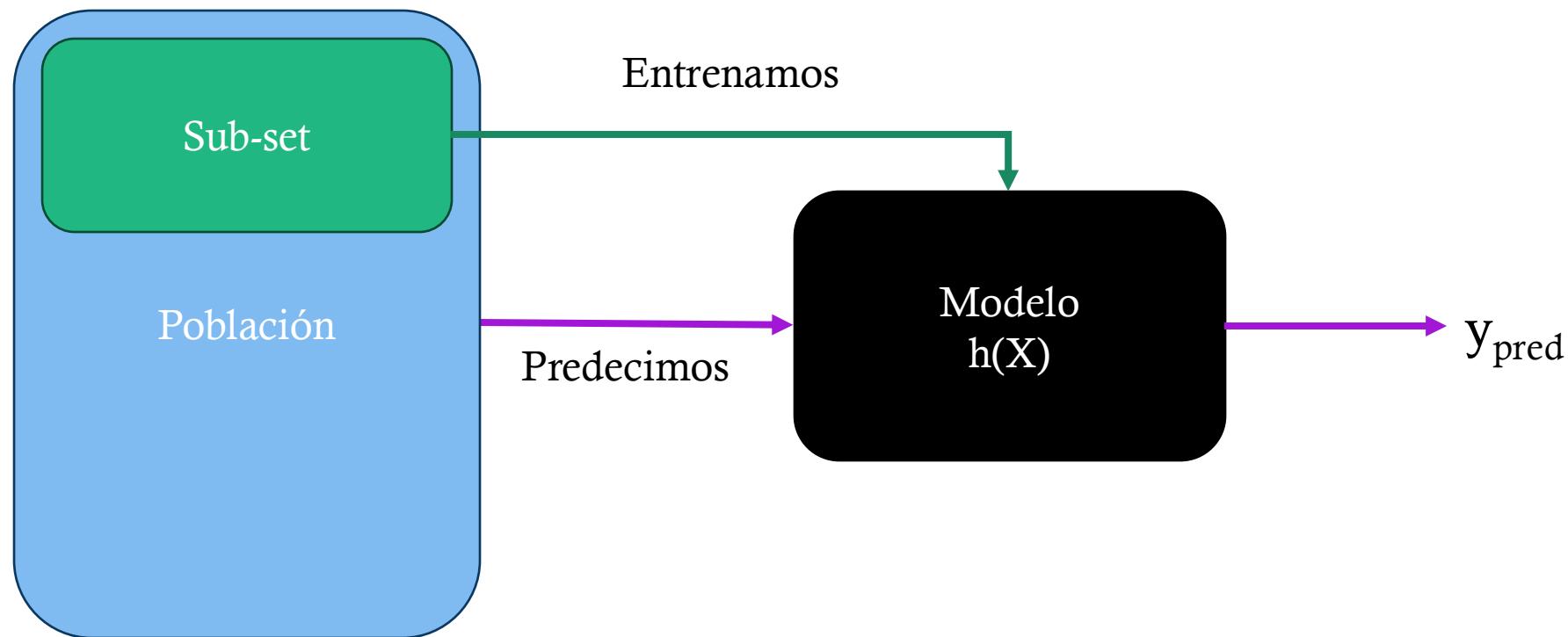
Ordinal



# FORMAS DE APRENDIZAJE

# FORMAS DE APRENDIZAJE

Un esquema de aplicar aprendizaje automático nos queda...



# FORMAS DE APRENDIZAJE

Como vimos en los ejemplos, hay diferentes tipos de aprendizaje, los cuales depende de la forma que tiene principalmente  $y$ .

- **Aprendizaje supervisado:** El modelo observa pares de entradas-salidas y aprende la relación entre ellos. Es decir, en este tipo de aprendizaje, conocemos el valor de  $y$  y se lo enseñamos al modelo.
  - Los modelos aprenden de los resultados conocidos y realizan ajustes en sus parámetros internos para adaptarse a los datos de entrada.
  - Una vez que el modelo está entrenado adecuadamente y los parámetros internos son coherentes con los datos de entrada y los resultados de los datos de entrenamiento, el modelo podrá realizar predicciones adecuadas ante nuevos datos.



---

# FORMAS DE APRENDIZAJE

Como vimos en los ejemplos hay diferentes tipos de aprendizaje, los cuales depende de la forma que tiene principalmente **y**.

- **Aprendizaje no supervisado:** El modelo aprende patrones de la entrada sin ninguna retroalimentación. Es decir, no contamos con **y** de antemano.
- **Aprendizaje por refuerzo:** El agente aprende con una serie de refuerzos: recompensas y castigos. Depende del agente decidir cuál de las acciones anteriores al refuerzo fue la más responsable de él y modificar sus acciones para apuntar a más recompensas en el futuro.

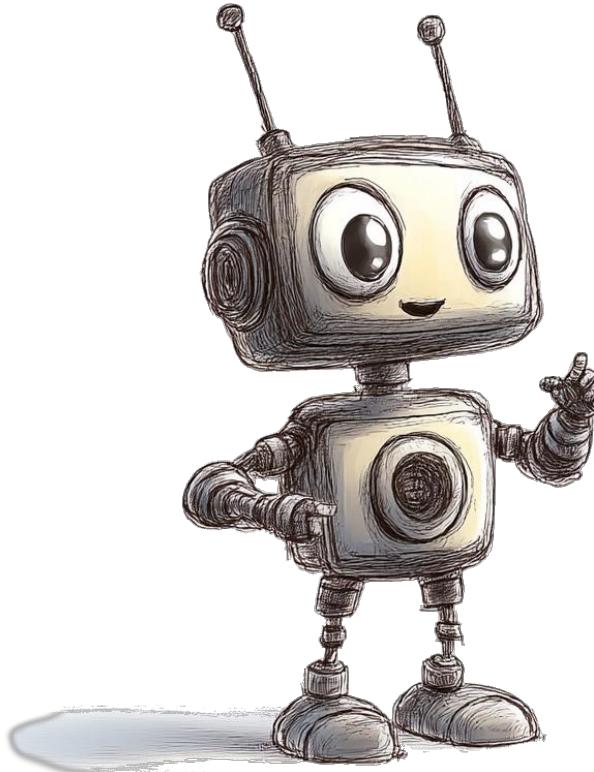
# APRENDIZAJE SUPERVISADO



Inteligencia Artificial  
CEIA - FIUBA  
Dr. Ing. Facundo Adrián  
Lucianna

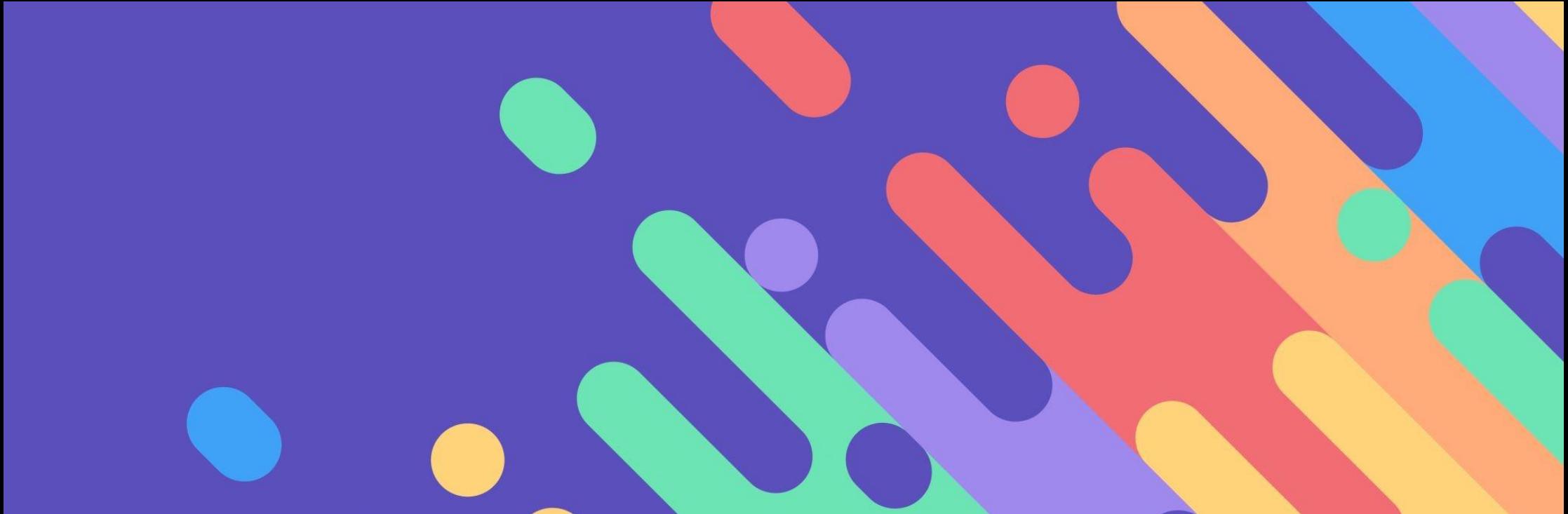
---

# APRENDIZAJE SUPERVISADO



Los temas que vamos a ver en este video son:

- Aprendizaje supervisado
  - Problema de clasificación
  - Problema de regresión
- Sesgo y varianza
- Estrategias para reducir el riesgo empírico:
  - Métodos de validación
  - Regularización



---

# APRENDIZAJE SUPERVISADO

---

# APRENDIZAJE SUPERVISADO

Más formalmente, podemos definir la tarea de aprendizaje supervisado como:

Dado un **set de entrenamiento** de N observaciones de pares de entradas y salida:

$$(X_1, y_1), (X_2, y_2), \dots, (X_N, y_N)$$

donde cada par está generado por una función desconocida  $y = f(X)$ ,

Se utiliza una función **h** que aproxima a la verdadera función  $f$ ,  $h(X) \approx f(X)$ .

La función **h** es la que llamamos la hipótesis de la población o el **modelo**, que proviene de un espacio de hipótesis.

Llamamos a las salidas **y** como **ground truth**.

---

# APRENDIZAJE SUPERVISADO

## *¿Cómo elegimos el espacio de hipótesis?*

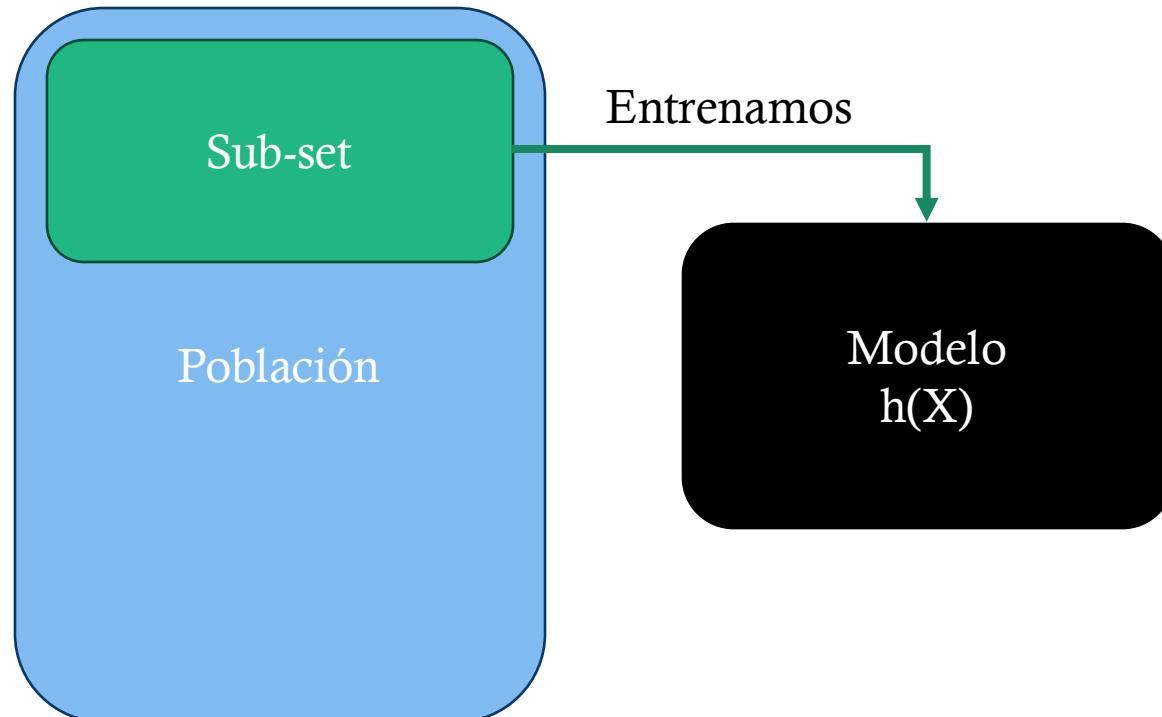
Es posible que tengamos algún conocimiento previo sobre el proceso que generó los datos, o podemos realizar un **análisis de datos exploratorio**: examinar los datos con pruebas estadísticas y visualizaciones para tener una idea de los datos y una noción de qué espacio de hipótesis podría ser apropiado. O simplemente podemos probar múltiples espacios de hipótesis y evaluar cuál funciona mejor.

## *¿Y cómo elegimos un modelo dentro del espacio de hipótesis?*

Podríamos esperar una hipótesis consistente: una **h** tal que cada  $X$  en el conjunto de entrenamiento tenga  $h(x) = y$ . Si las salidas son de valor continuo, es muy difícil tener una salida exacta. En esos casos, se usa una **función de ajuste** para medir que tan  $h(x_i)$  está cerca de  $y_i$ .

---

# APRENDIZAJE SUPERVISADO

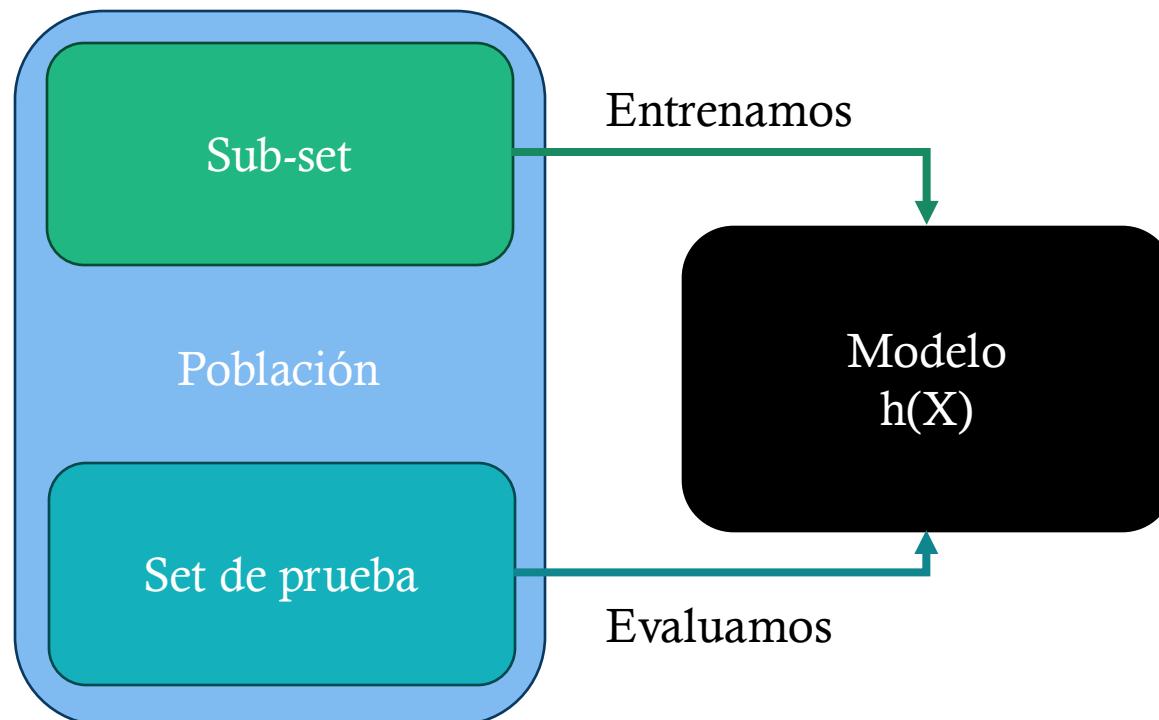


En la realidad, no podemos saber exactamente qué tan bien funcionará un modelo predictivo en la práctica porque no conocemos la verdadera distribución de los datos.

Siempre vamos a trabajar con subconjuntos que podemos estimar y optimizar el rendimiento del modelo en un conjunto conocido de datos de entrenamiento.

El rendimiento sobre este conjunto conocido de datos de entrenamiento se denomina **riesgo empírico**.

# APRENDIZAJE SUPERVISADO



## Generalización

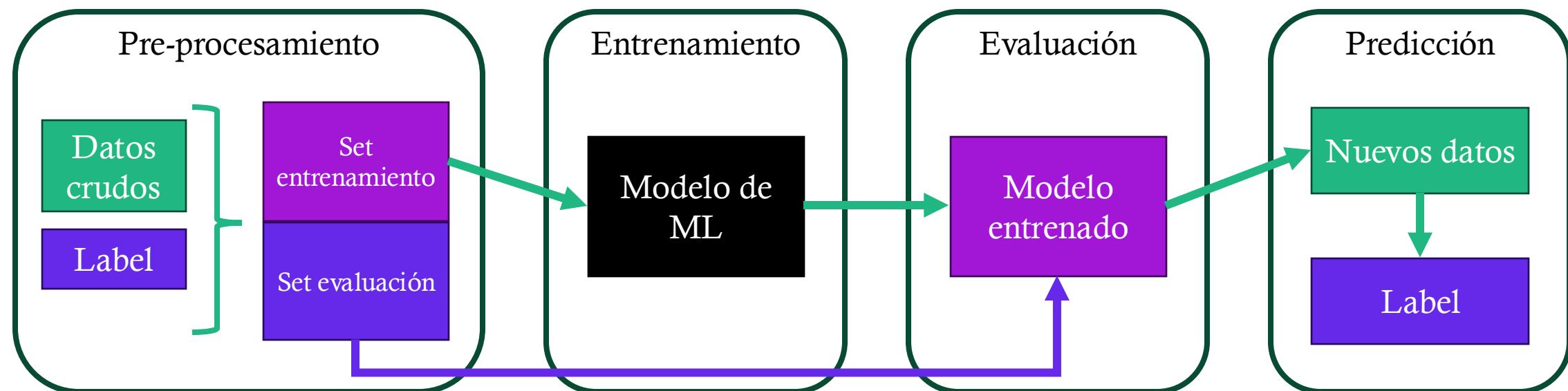
La verdadera medida de si un modelo funciona correctamente no es con respecto al conjunto de entrenamiento, sino qué tan bien maneja entradas que nunca vio.

Eso lo podemos ver con un segundo set de pares  $(X_i, y_i)$  llamada **conjunto de prueba**.

Se dice que  $h$  **generaliza** bien si predice con precisión los resultados de este conjunto.

# APRENDIZAJE SUPERVISADO

## Generalización

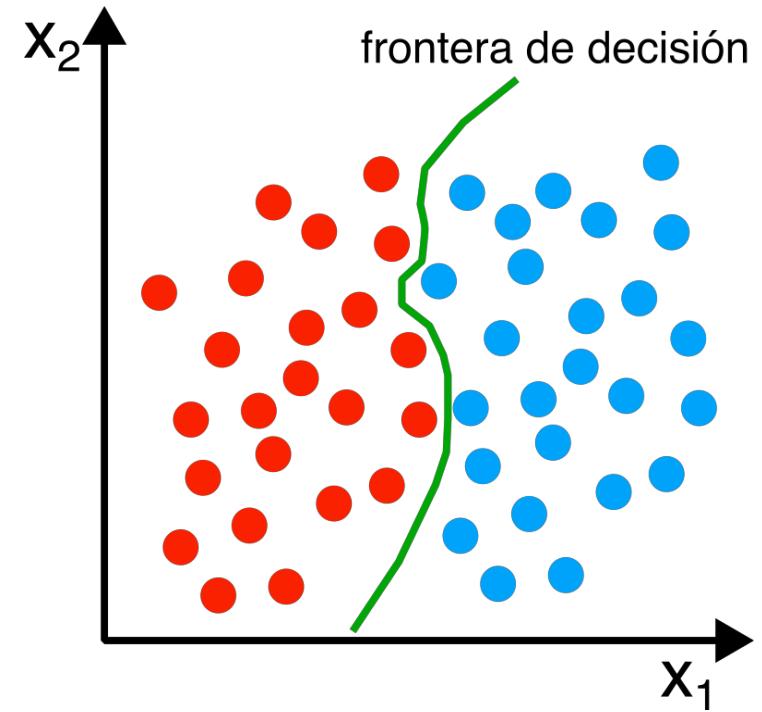


# APRENDIZAJE SUPERVISADO

## Tipos de aprendizaje supervisado

Si el target **y** es una *variable categórica* o toma valores discretos, este tipo de problema se llama un **problema de clasificación**. A su vez, tenemos dos subvariantes:

- **Clasificación binaria** (por ejemplo, es SPAM o no SPAM).
- **Clasificación multi-clase:** múltiples clases, como por ejemplo, la clasificación del nivel socioeconómico de una persona (alta, media y baja).
  - Una variante de este tipo es cuando la cardinalidad es muy alta, es decir, se tienen muchísimas clases.

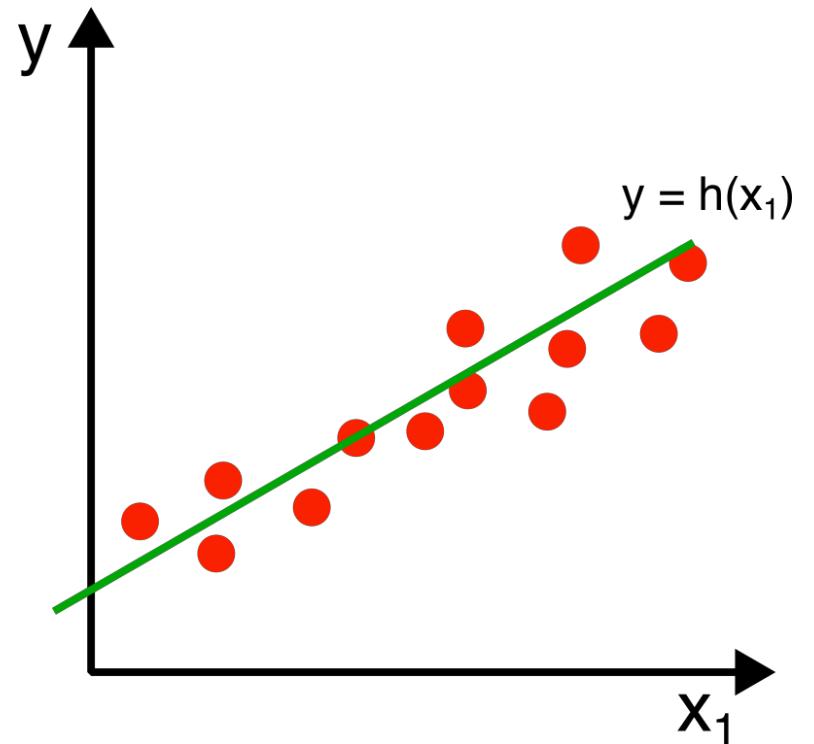


# APRENDIZAJE SUPERVISADO

## Tipos de aprendizaje supervisado

Si el target **y** es una *variable numérica*, este tipo de problema se llama un **problema de regresión**.

Se centra en estudiar las relaciones entre una variable dependiente de una o más variables independientes.



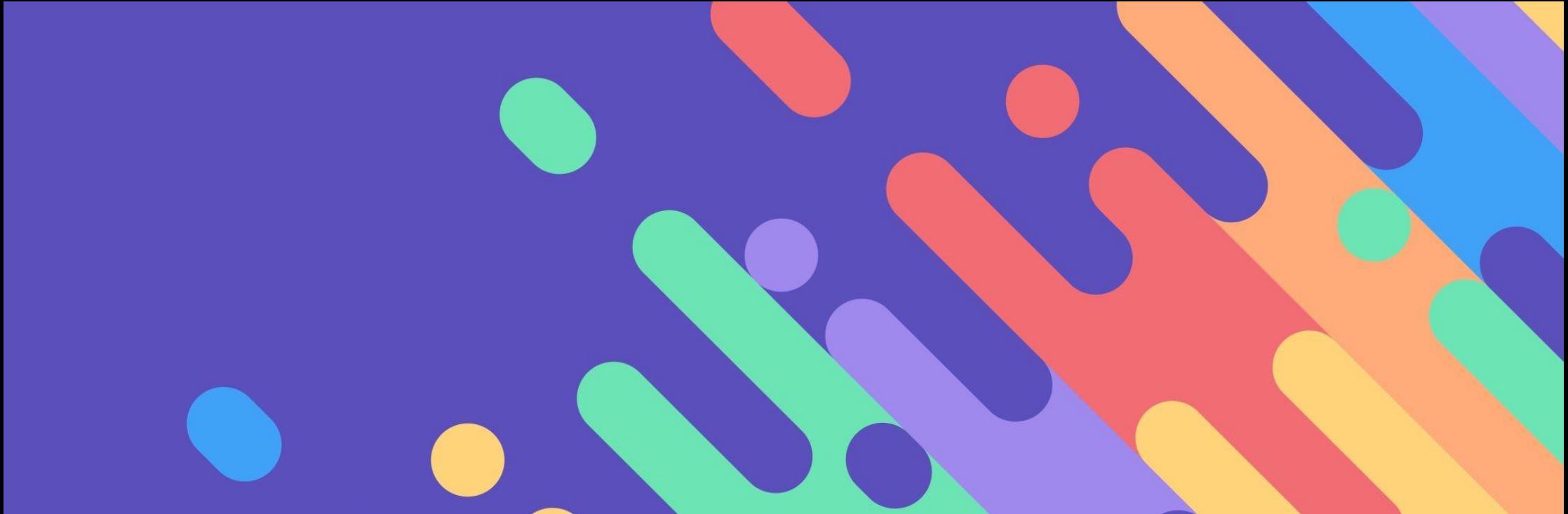
---

# APRENDIZAJE SUPERVISADO

Tipos de aprendizaje supervisado

## Regresión vs. Clasificación

Regresión y clasificación son problemas muy similares entre sí. En ambos buscamos predecir una variable; la diferencia radica en que la regresión predice una variable numérica y la clasificación una categórica.

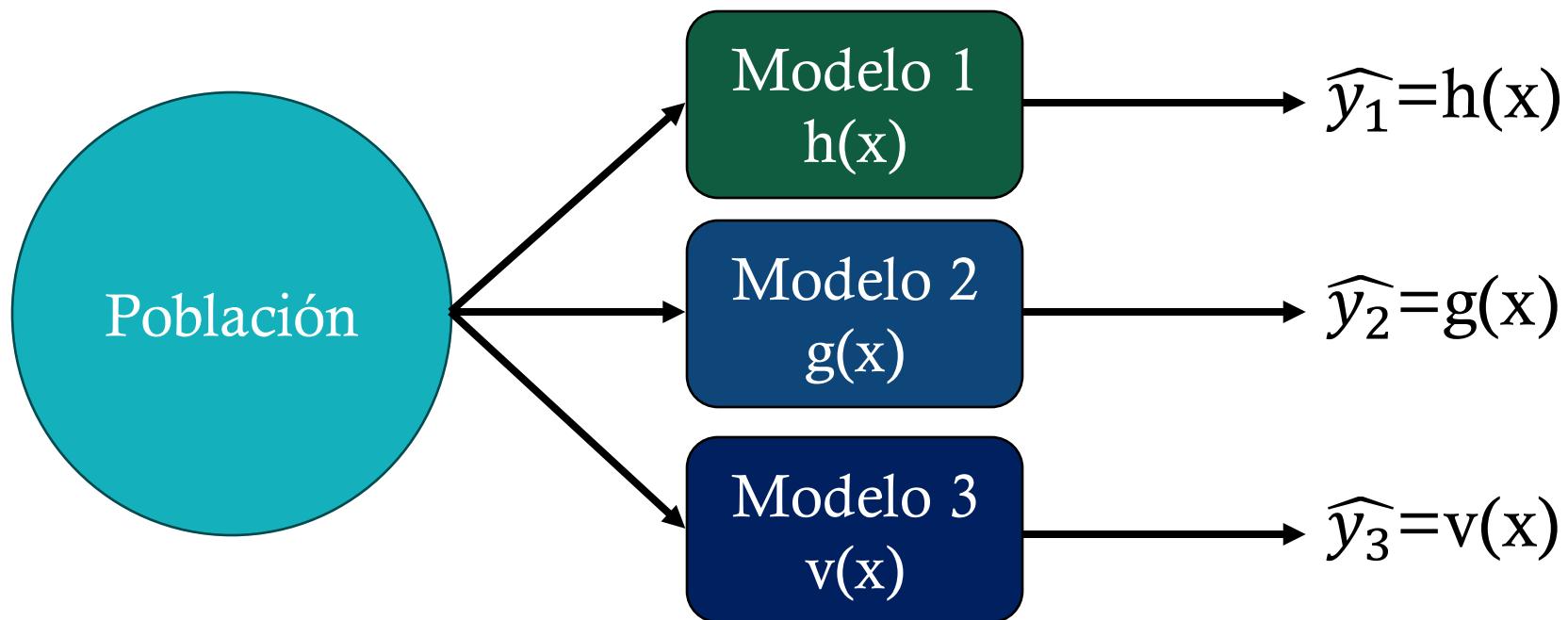


---

# SESGO Y VARIANZA

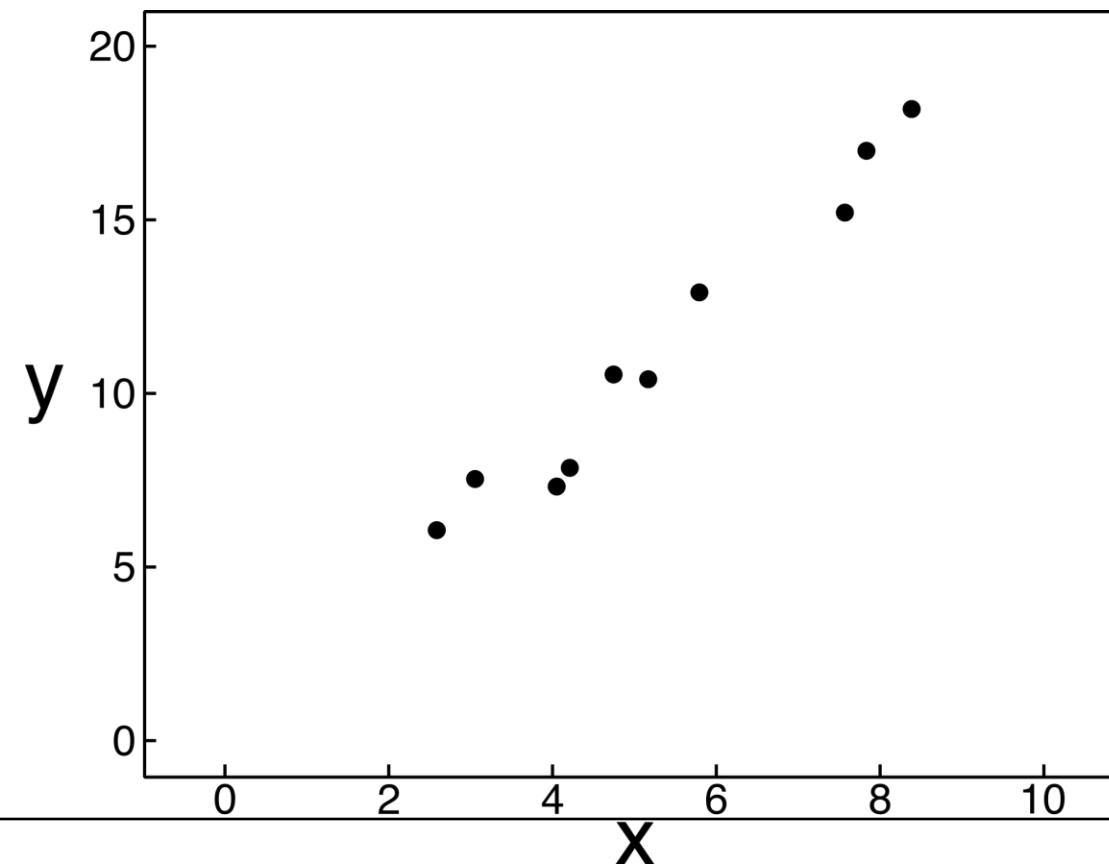
# SESGO Y VARIANZA

Supongamos que tenemos tres modelos en un problema de regresión. Tenemos una sola entrada  $x$  y una salida  $y$  que depende de  $x$  con una relación  $f(x)$  que queremos modelar:



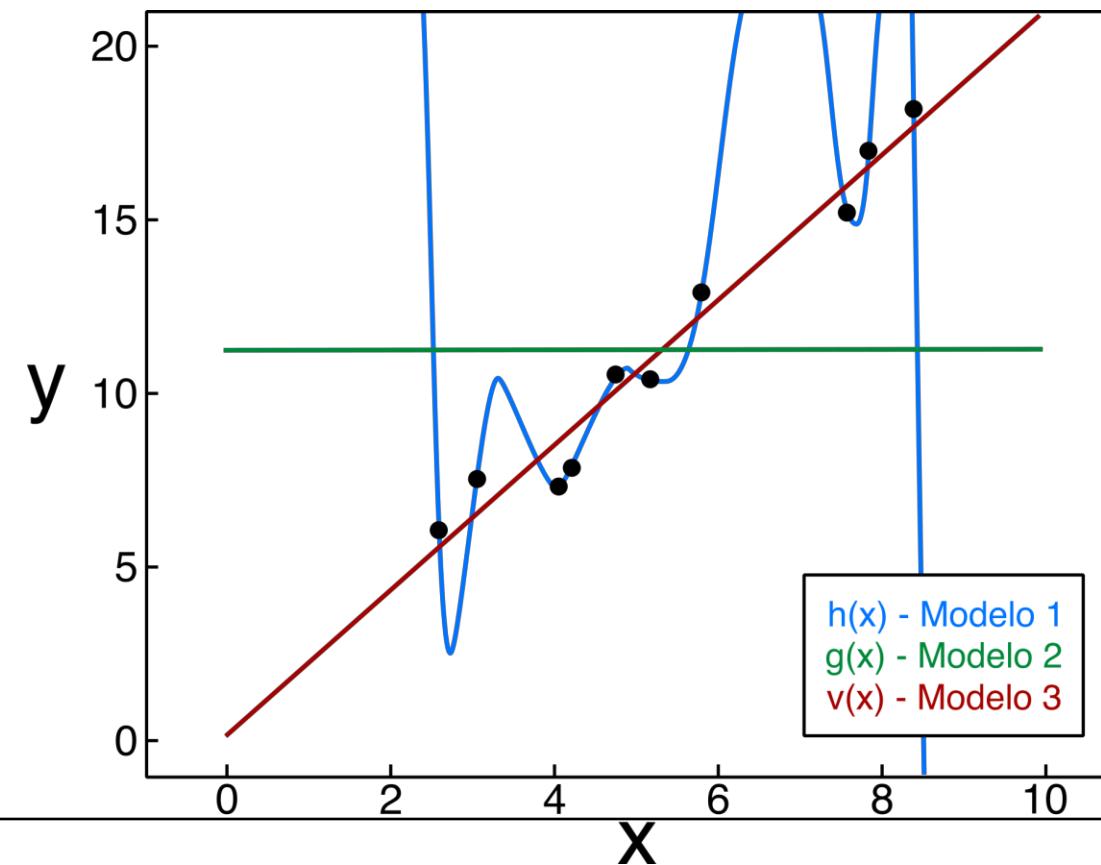
# SESGO Y VARIANZA

Entrenamos con un **set de entrenamiento** y nos queda:



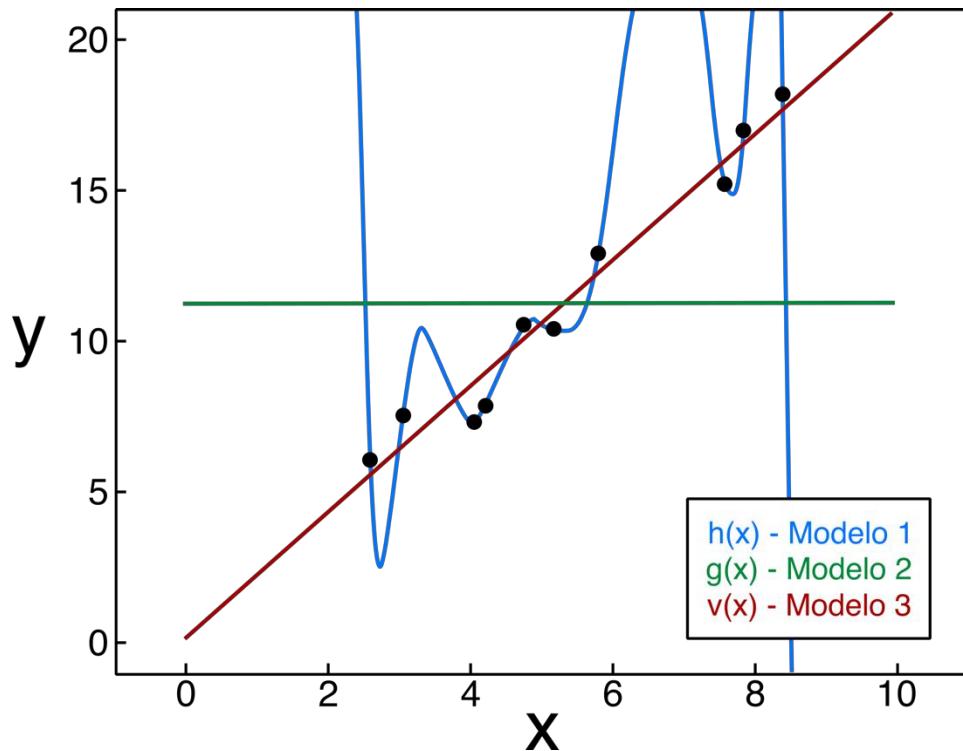
# SESGO Y VARIANZA

Entrenamos con un **set de entrenamiento** y nos queda:



# SESGO Y VARIANZA

Entrenamos con un **set de entrenamiento** y nos queda:



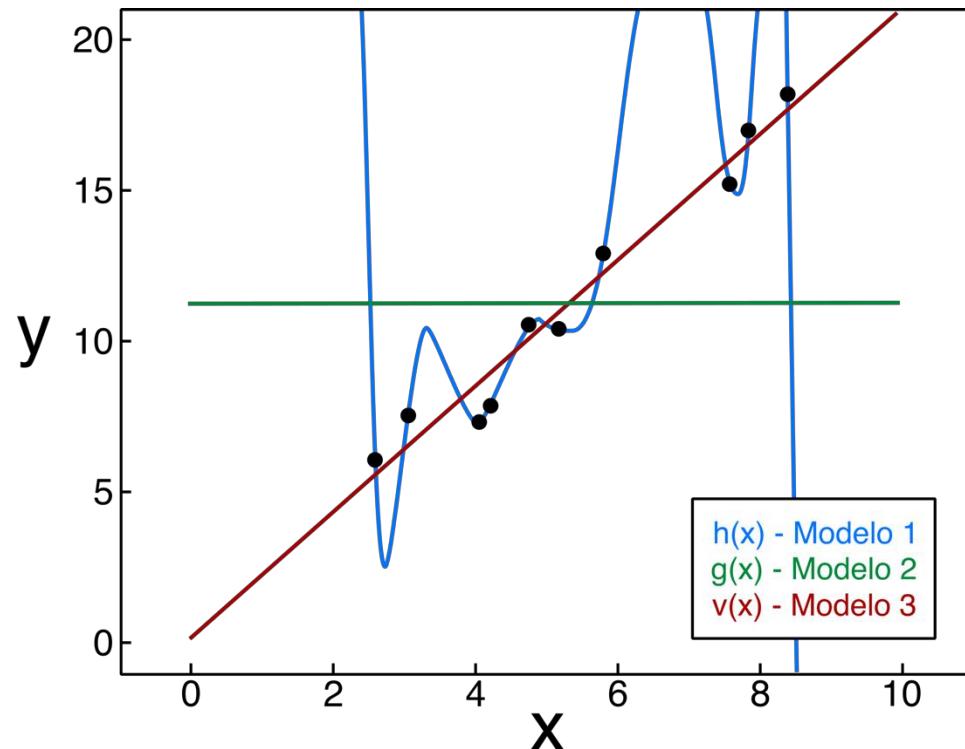
Midamos el error, para ello calculamos de la siguiente forma:

$$MAE_p = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_p|$$

- Modelo 1:  $MAE_1 = 0$
- Modelo 2:  $MAE_2 = 5$
- Modelo 3:  $MAE_3 = 1.25$

# SESGO Y VARIANZA

Entrenamos con un **set de entrenamiento** y nos queda:



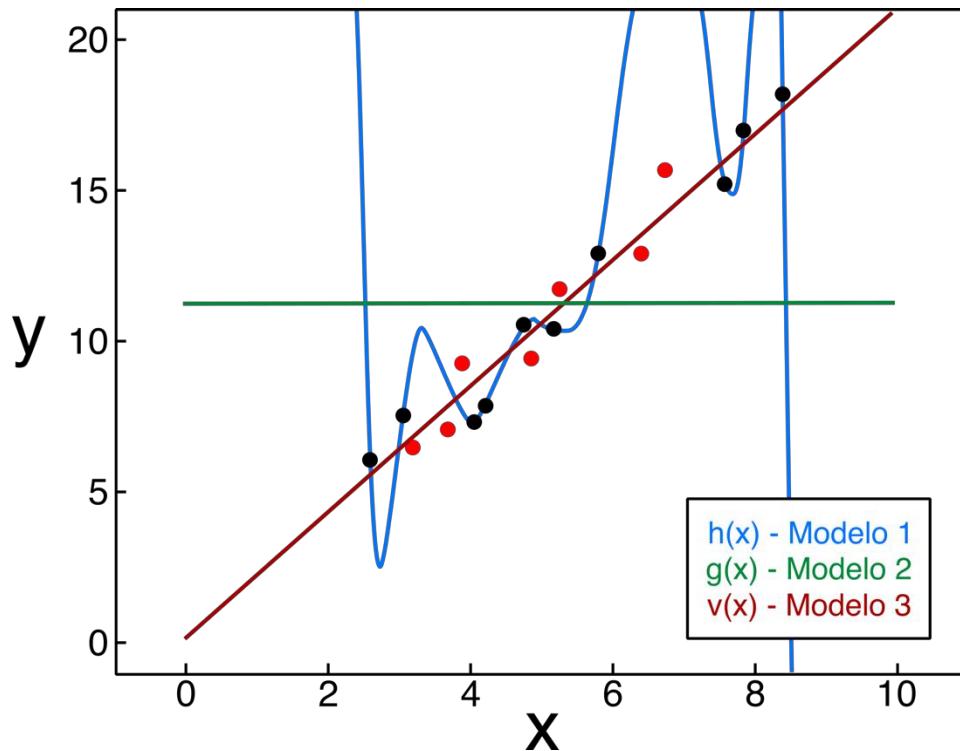
Midamos el error, para ello calculamos de la siguiente forma:

$$MAE_p = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_p|$$

- **Modelo 1:  $MAE_1 = 0$**
- **Modelo 2:  $MAE_2 = 5$**
- **Modelo 3:  $MAE_3 = 1.25$**

# SESGO Y VARIANZA

¿Pero realmente es el mejor? Evaluemos con el **set de evaluación**



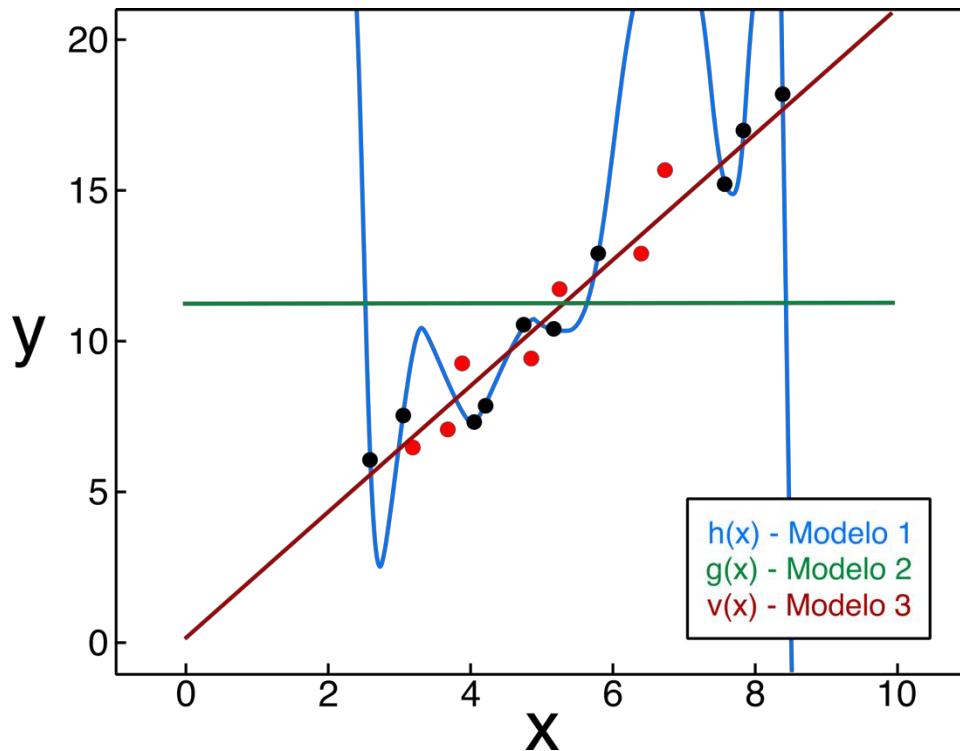
Midamos el error...

$$MAE_p = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_p|$$

- Modelo 1:  $MAE_1 = 90$
- Modelo 2:  $MAE_2 = 5.2$
- Modelo 3:  $MAE_3 = 1.75$

# SESGO Y VARIANZA

¿Pero realmente es el mejor? Evaluemos con el **set de evaluación**



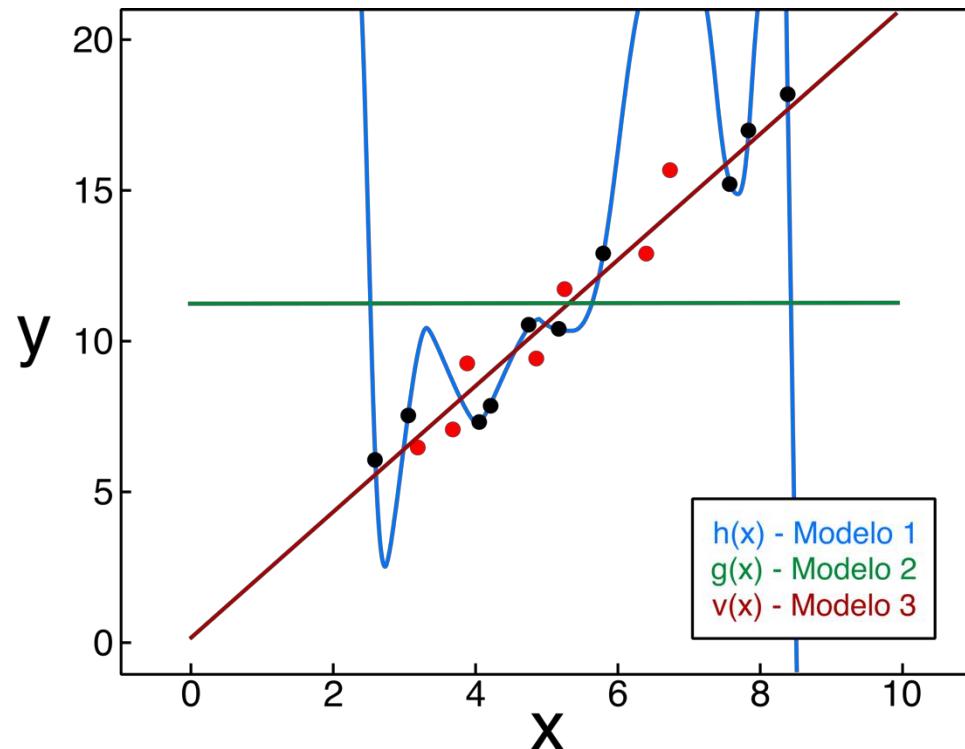
Midamos el error...

$$MAE_p = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_p|$$

- Modelo 1:  $MAE_1 = 90$
- Modelo 2:  $MAE_2 = 5.2$
- **Modelo 3:  $MAE_3 = 1.75$**

# SESGO Y VARIANZA

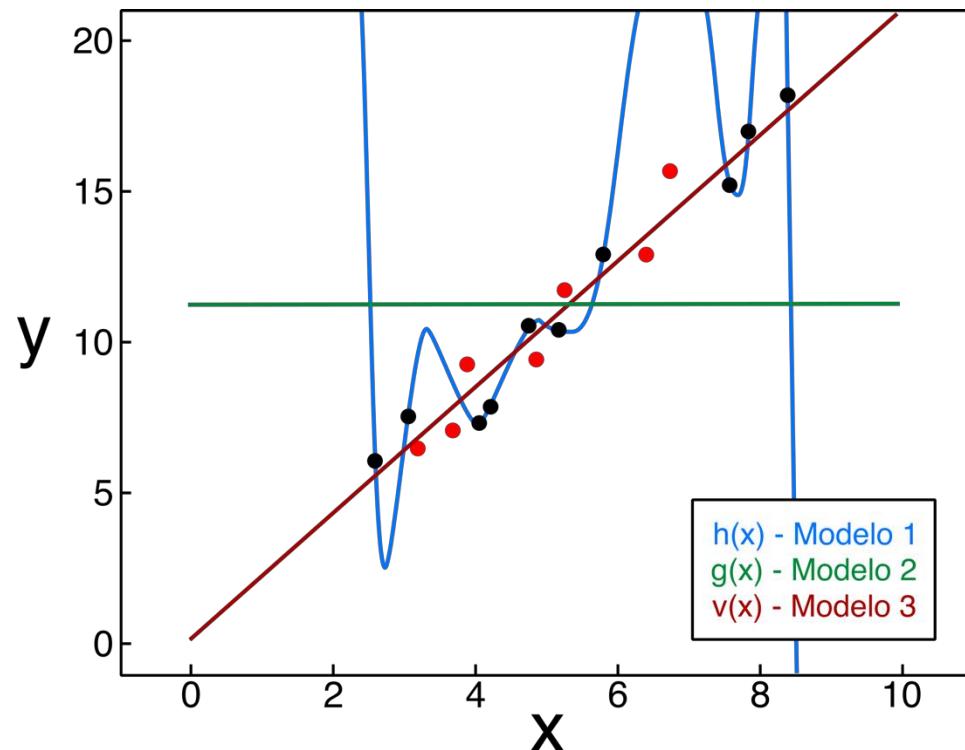
¿Pero realmente es el mejor? Evaluemos con el **set de evaluación**



- El modelo 1 rinde muy bien con los datos de entrenamiento, pero muy mal con los de evaluación. Esto es lo que se llama **sobreajuste**: el modelo aprende tan bien que incluso aprende el ruido de los datos.  
Este modelo tiene demasiados parámetros, por lo que es demasiado complejo y, por consiguiente, **no generaliza**.
- Cuando se elige un modelo, se busca que:  
$$\text{parametros} \ll \text{cantidad de obs. de entrenamiento}$$

# SESGO Y VARIANZA

¿Pero realmente es el mejor? Evaluemos con el **set de evaluación**



- El modelo 2 rinde pobemente tanto con el conjunto de entrenamiento como con el de evaluación. Este modelo no logra capturar el comportamiento buscado. Cuando esto ocurre, lo que llamamos es **subajuste**.
- El modelo 3 es el mejor modelo; el resultado entre el conjunto de evaluación y el de entrenamiento es similar, el modelo **generaliza**, sin tomar el ruido.

---

# SESGO Y VARIANZA

El error que se comete con un modelo que sobreajusta y subajusta es diferente.

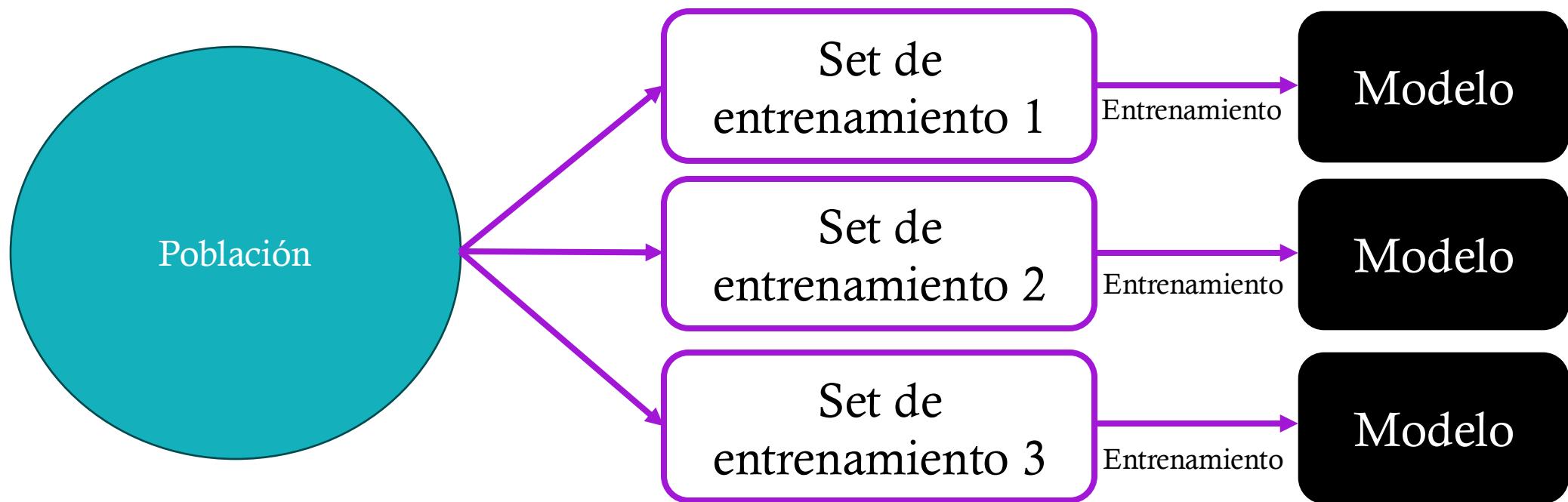
- El error de **sobreajuste** lo llamamos **error de varianza**
- El error de **subajuste** lo llamamos **error de sesgo**

Todo modelo tiene un error total, que está compuesto por ambos errores.

Estos errores son complementarios entre sí. Si queremos reducir uno, el otro va a aumentar.

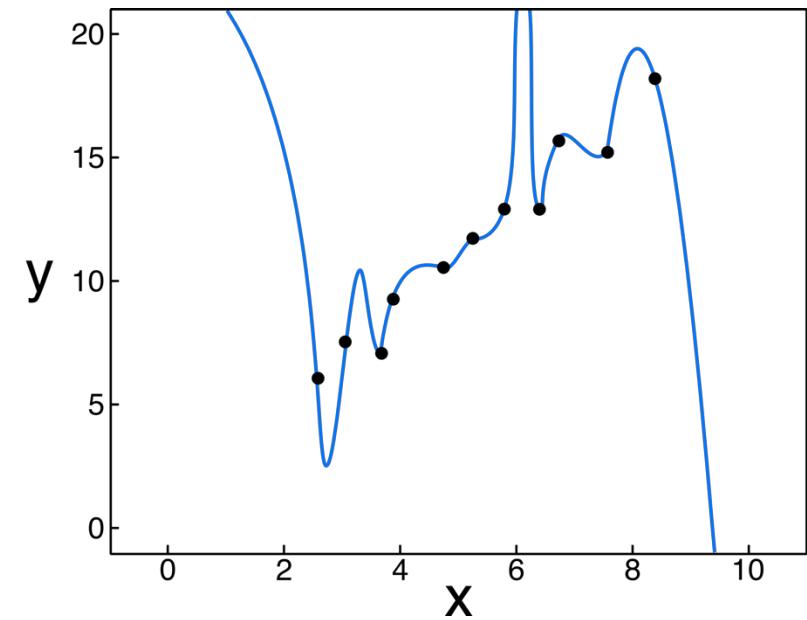
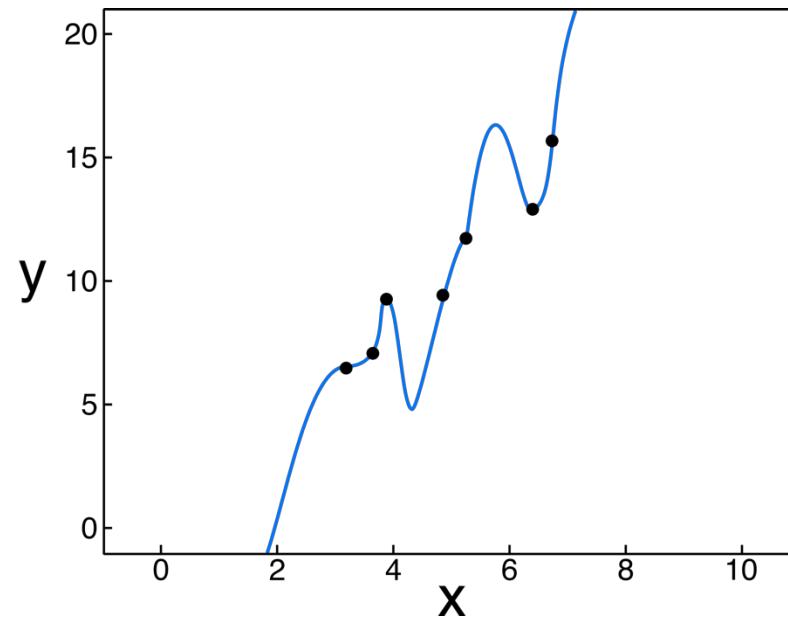
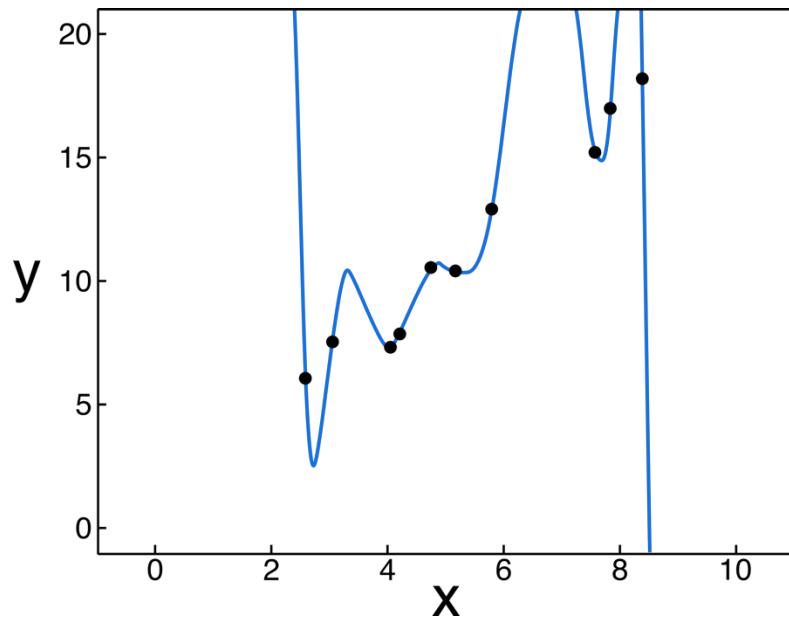
# SESGO Y VARIANZA

Estos errores los podemos ver si tomamos diferentes *sets de entrenamiento* y entrenamos el modelo con cada uno de ellos.



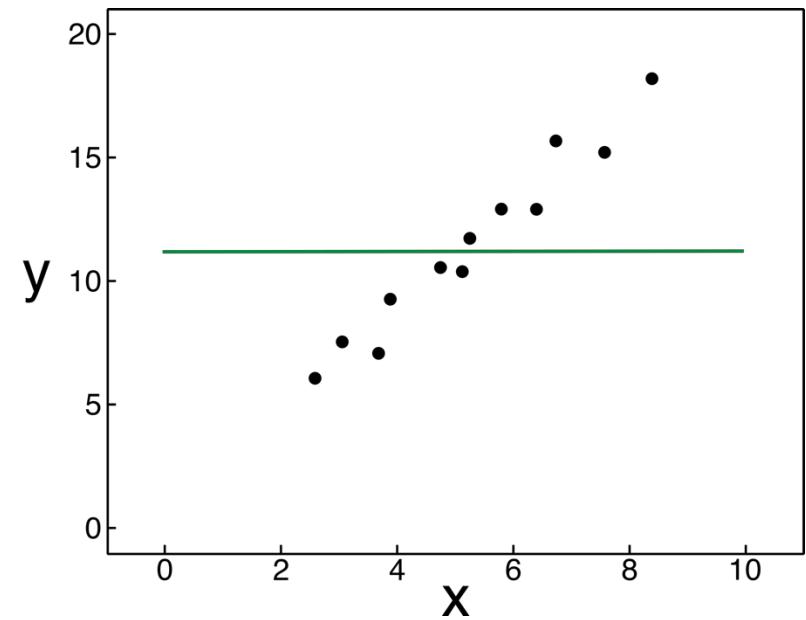
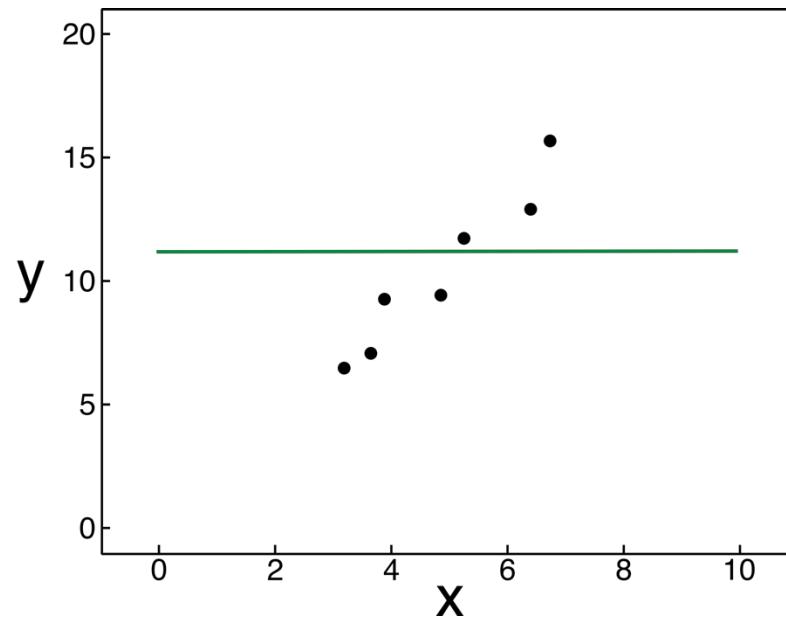
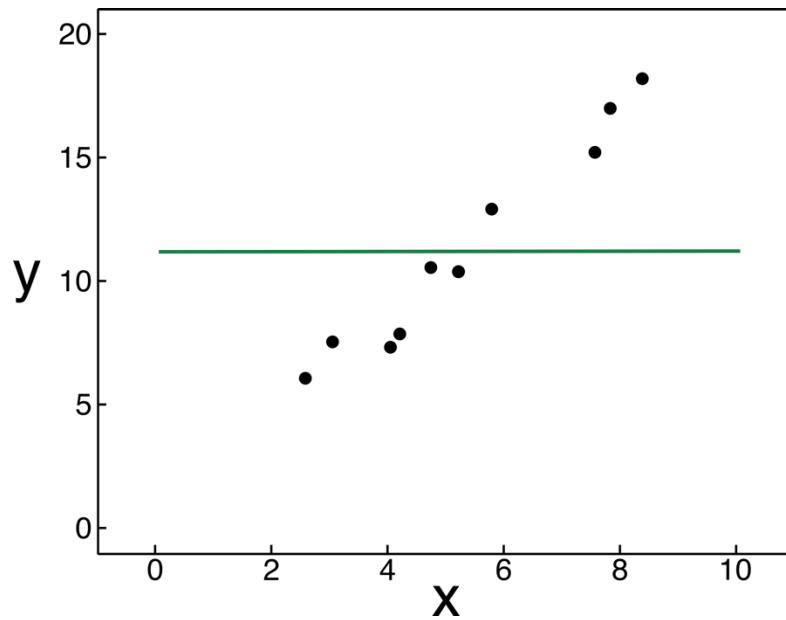
# SESGO Y VARIANZA

Si tenemos un modelo que **sobreajusta**, diferentes sets de entrenamiento nos darán modelos muy diferentes:



# SESGO Y VARIANZA

Si tenemos un modelo que **subajusta**, diferentes sets de entrenamiento nos darán modelos muy similares:



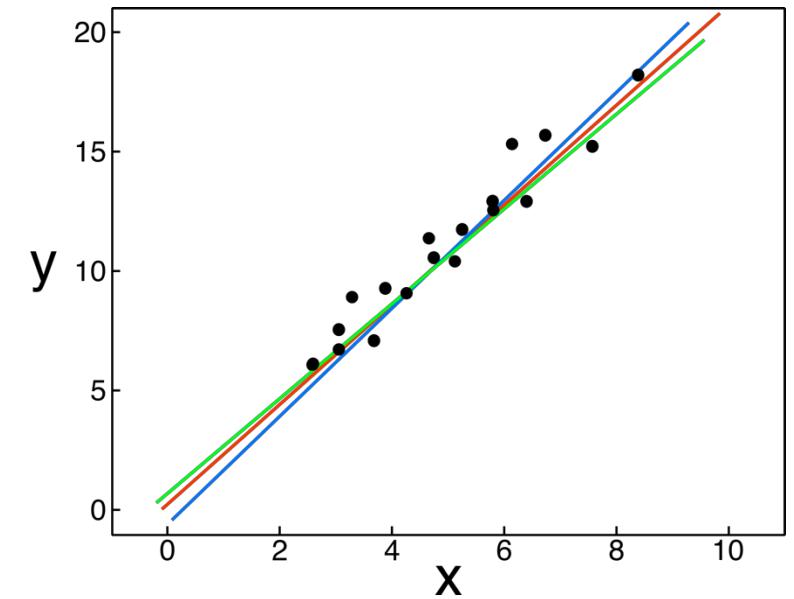
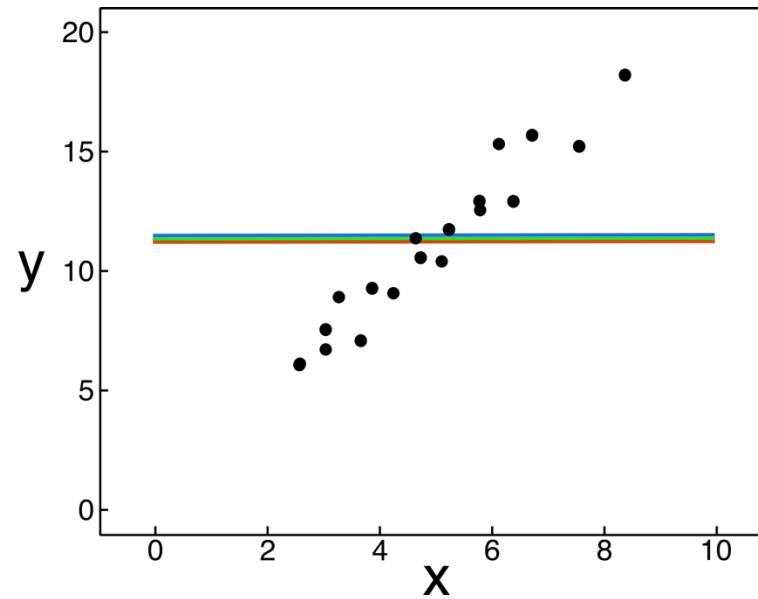
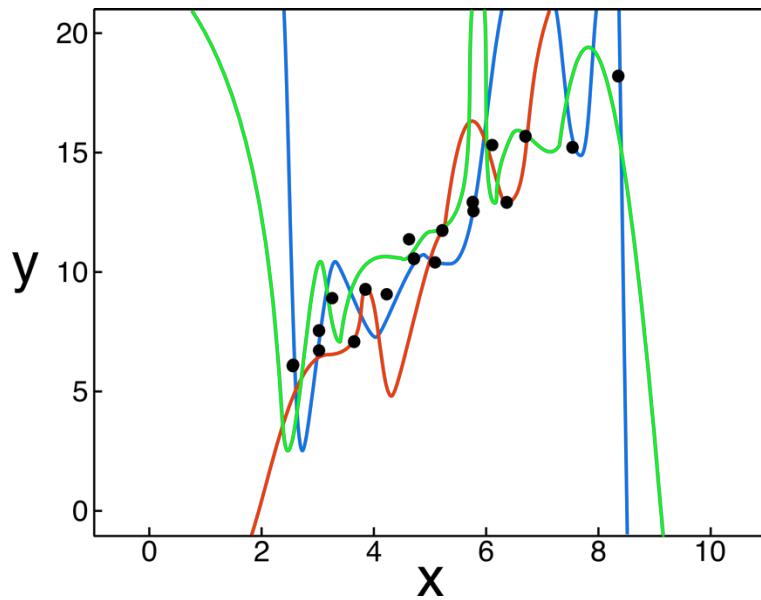
---

# SESGO Y VARIANZA

- El modelo más simple cometerá muchos errores en prácticamente cualquier conjunto de entrenamiento, lo que significa que tiene **un error alto de sesgo**. Sin embargo, cuando evaluamos estos modelos con el conjunto de evaluación, el resultado siempre es similar. Entonces, decimos que tiene **un error de varianza bajo**.
- Por otro lado, el modelo complejo se adapta perfectamente a cada conjunto de entrenamiento. Tiene **un error de sesgo muy bajo**, pero un **error de varianza muy alto** (ya que dos conjuntos de entrenamiento cualesquiera probablemente darían lugar a modelos muy diferentes).

# SESGO Y VARIANZA

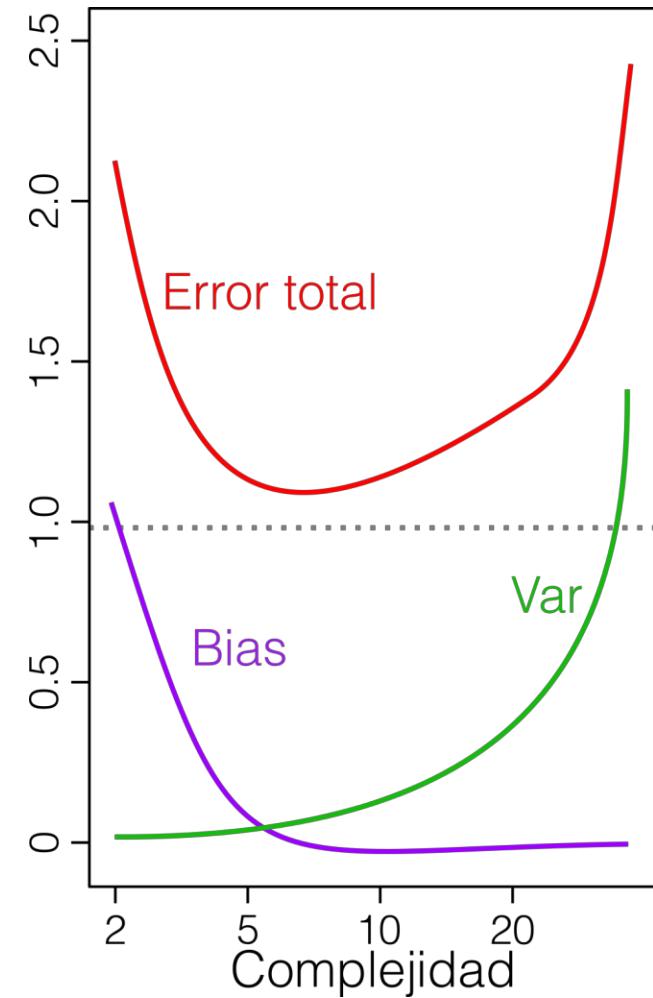
- Si analizamos cada caso, observamos que el modelo lineal, aunque tiene varianza, el error es pequeño, ya que tiene poca varianza y poco sesgo.

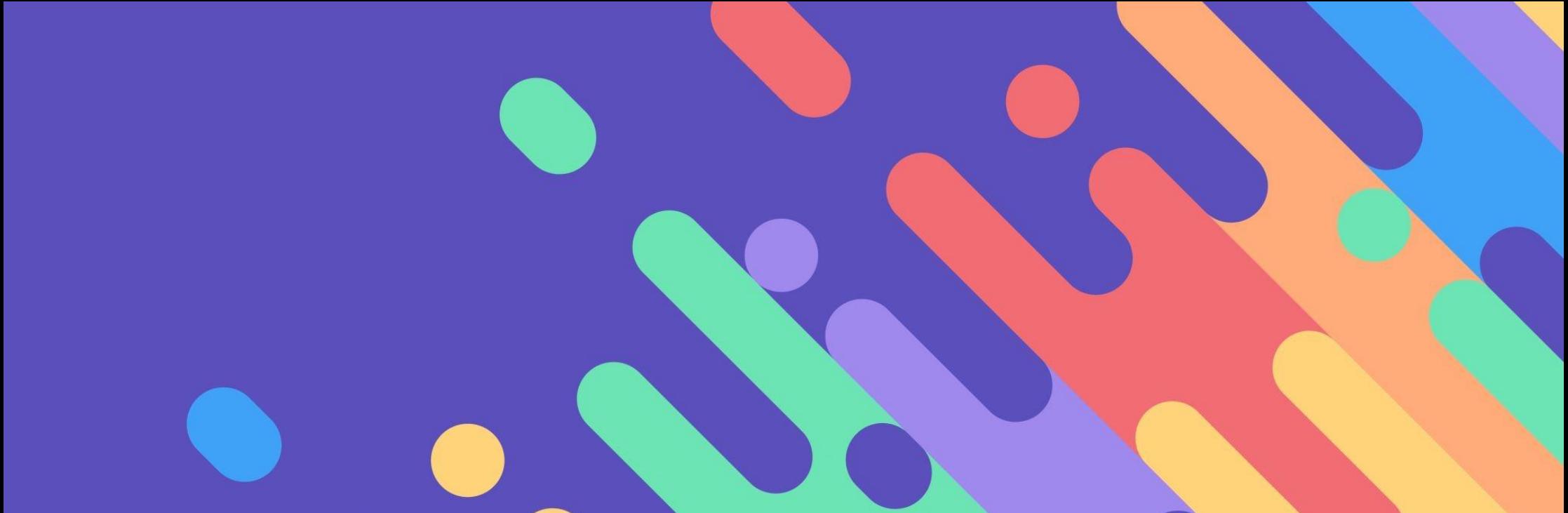


# SESGO Y VARIANZA

*Como regla general,*

- Cuando más complejo es el modelo, la varianza aumentará y el sesgo disminuirá
- Cuando aumentamos la complejidad de este, el sesgo tiende a disminuir más rápido de lo que la variabilidad aumenta, reduciendo el error.
- Llega un punto en el que el efecto de la variabilidad es apreciable, aumentando el valor del error nuevamente.





---

# ESTRATEGIAS PARA DISMINUIR EL RIESGO EMPÍRICO

---

# ESTRATEGIAS

Pensemos ahora en los modelos como cajas grises; todavía no sabemos cómo funcionan, pero tenemos una idea de que tienen parámetros, los cuales son de dos tipos:

- **Parámetros que se entrena**n: Son los parámetros que un modelo aprende cuando lo entrenamos. Ya vimos que su número debe ser mucho menor que la cantidad de datos de entrenamiento para evitar el sobreajuste.
- **Hiperparámetros**: Son parámetros *internos* que no dependen de los datos. Estos deben ser definidos antes de entrenar.

# ESTRATEGIAS



Elegir estos hiperparámetros es un desafío importante, ya que, para tener el mejor modelo, deberíamos saber de antemano, pero para saberlo, debemos entrenar.

Entonces, lo que se hace es entrenar muchos modelos con diferentes hiperparámetros, y ver cuál es el mejor.

¡Pero si usamos al **set de evaluación**, encontraremos el mejor para este con el riesgo de que **no generalicemos!**!

---

# ESTRATEGIAS

Entonces, debemos usar un tercer conjunto de datos que sacamos del de entrenamiento, llamado **set de validación**.

- Este conjunto nos permite evaluar diferentes valores de hiperparámetros
- Ver si el modelo está sobreajustando o evaluar el modelo mientras aprende.

*En Deep Learning es normal usarlo para evaluar durante el entrenamiento. Este entrenamiento es con gradiente descendente, y es importante ver, cada cierta iteración, cómo evoluciona.*

**Con este conjunto evitamos usar el de evaluación y “hacer trampa”.**

---

# ESTRATEGIAS

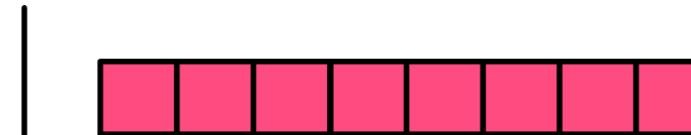
Esto mejora la generalización, pero es muy sensible a la selección del conjunto de validación. Además, quitamos datos que nos podrían haber servido para entrenar.

Una estrategia que nos evita esto es usar **validación cruzada**.

# ESTRATEGIAS

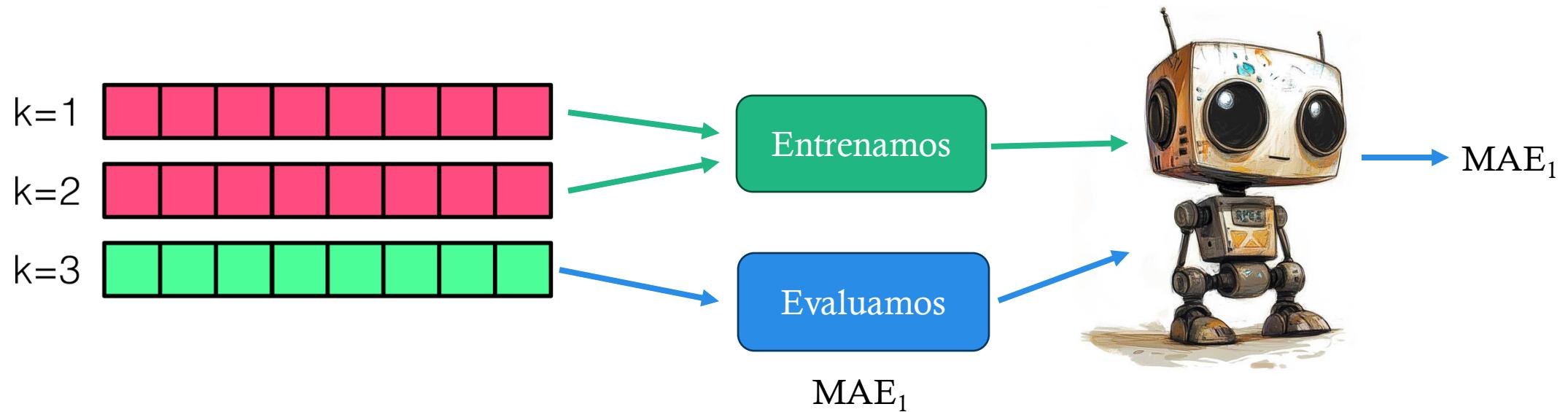
## Validación cruzada

La más conocida es **validación cruzada K-Fold**.



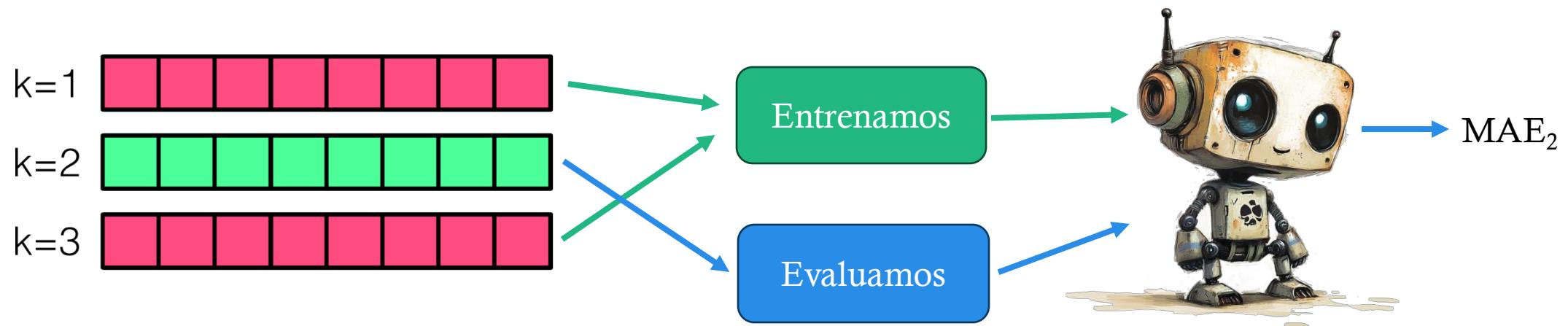
# ESTRATEGIAS

## Validación cruzada



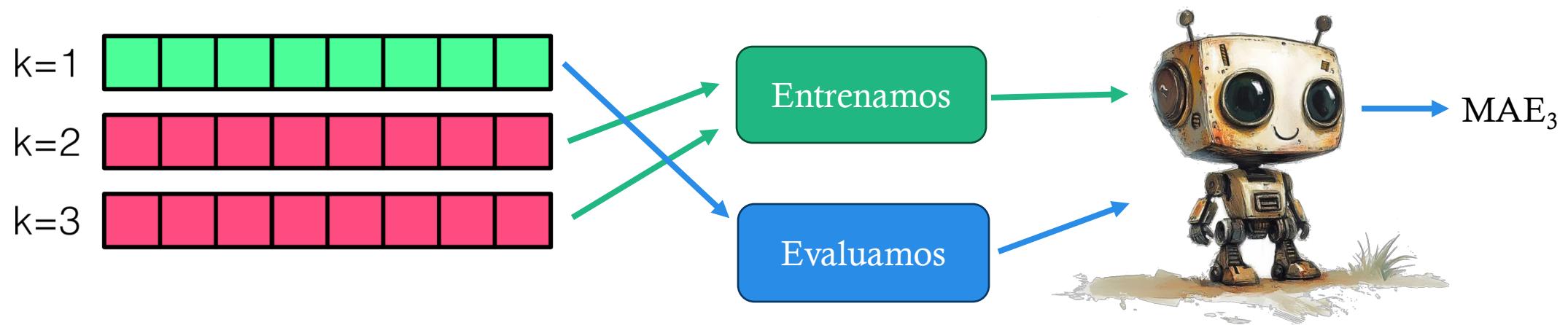
# ESTRATEGIAS

## Validación cruzada



# ESTRATEGIAS

## Validación cruzada

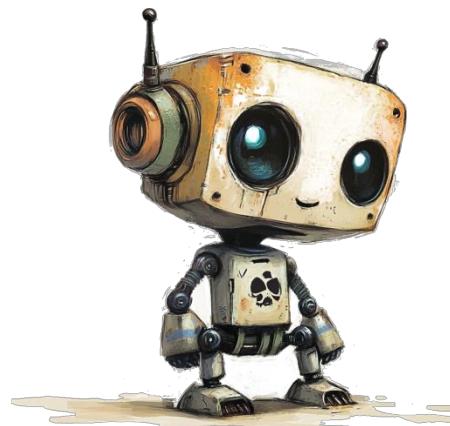


# ESTRATEGIAS

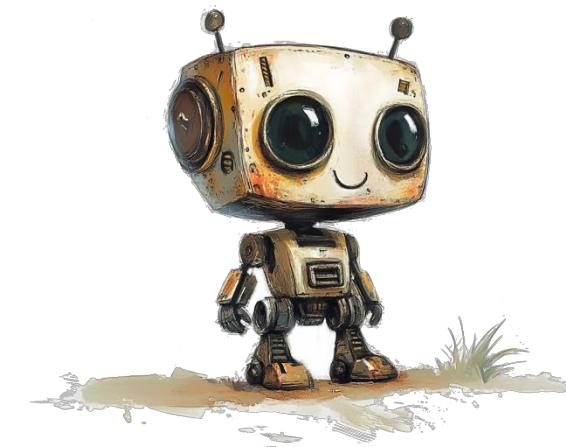
Validación cruzada



MAE<sub>1</sub>



MAE<sub>2</sub>



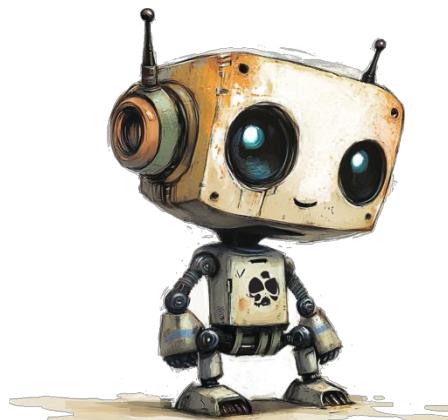
MAE<sub>3</sub>

# ESTRATEGIAS

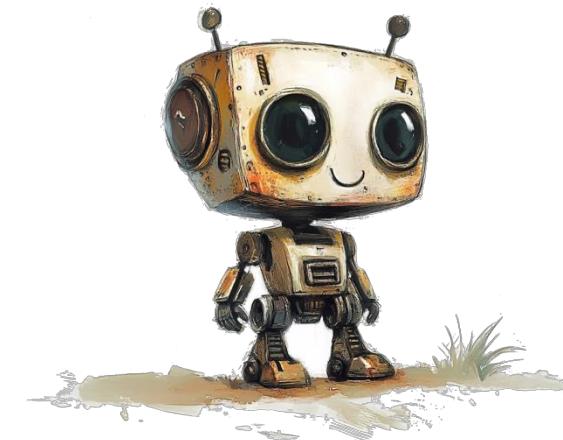
Validación cruzada



MAE<sub>1</sub>



MAE<sub>2</sub>

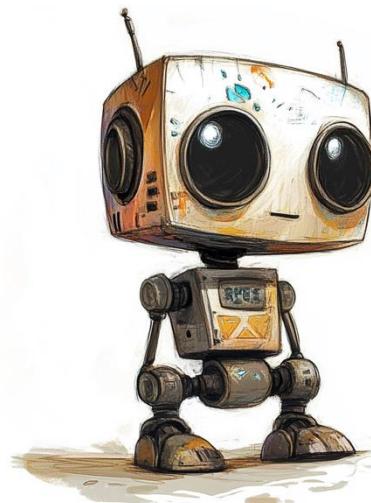


MAE<sub>3</sub>

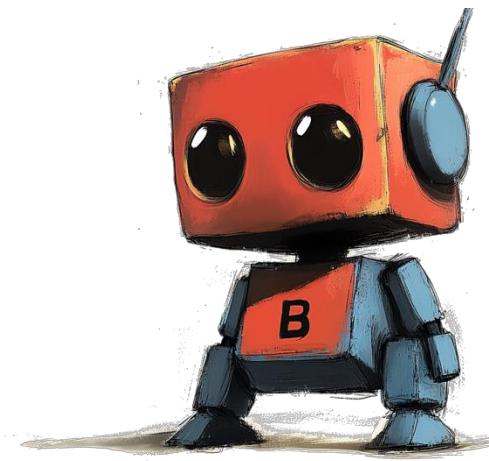
$\widetilde{MAE}$

# ESTRATEGIAS

Validación cruzada



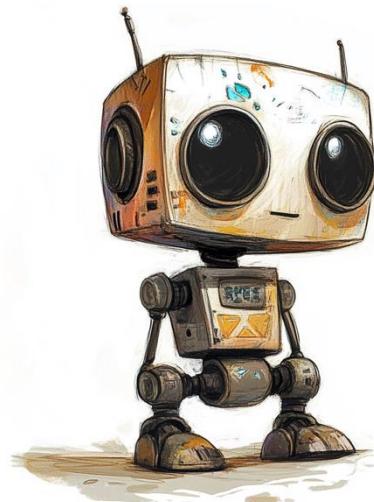
$\widehat{\text{MAE}}$



$\widehat{\text{MAE}}$

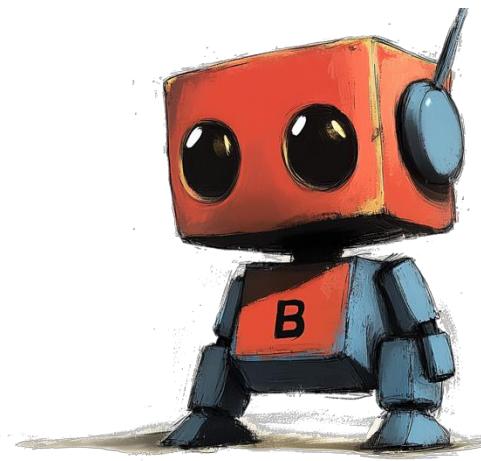
# ESTRATEGIAS

Validación cruzada



$\widetilde{\text{MAE}}$

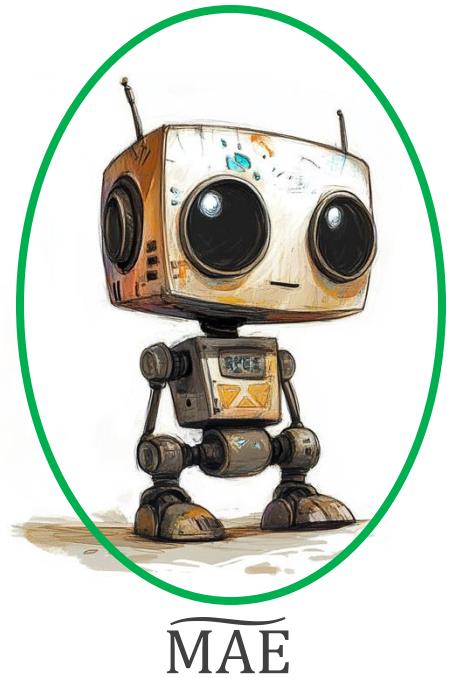
<



$\widetilde{\text{MAE}}$

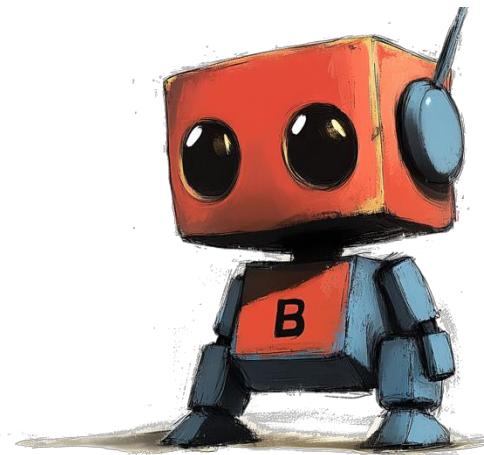
# ESTRATEGIAS

Validación cruzada



$\widetilde{MAE}$

<



$\widetilde{MAE}$

---

# ESTRATEGIAS

## Validación cruzada K-Fold.

### Beneficios:

- Estrategia sistemática para encontrar los hiperparámetros.
- Nos permite evaluar al menos una vez cada punto de entrenamiento.
- Nos permite evaluar problemas de sobreajuste.
- Sin tocar el conjunto de evaluación.

### Desventaja:

- Es un proceso muy lento. Cada modelo debe entrenarse K veces.

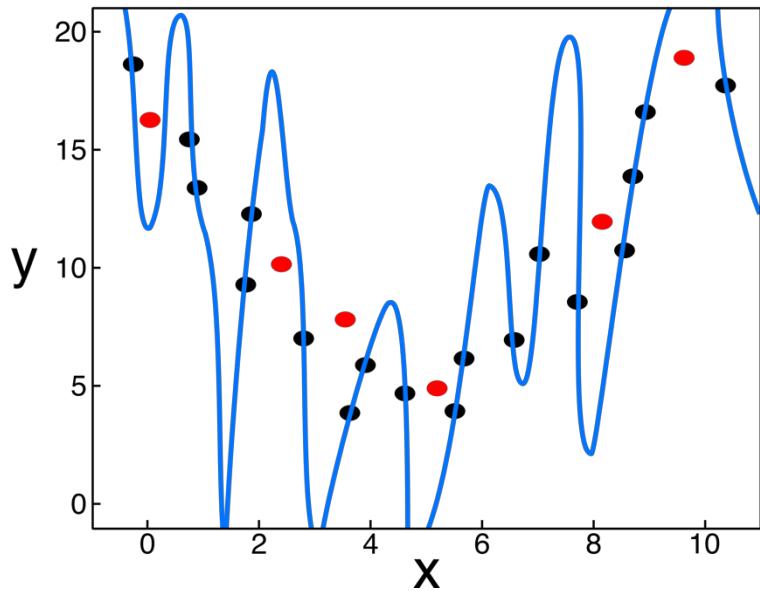
*En procesos lentos y con conjuntos de entrenamiento muy grandes, conviene usar un conjunto de validación.*

---

# ESTRATEGIAS

## Regularización

Sin regularizar

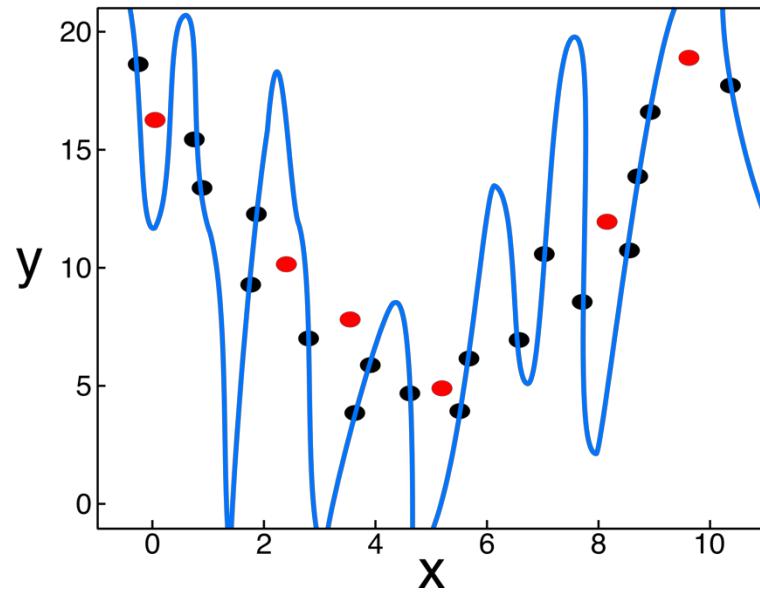


Error de varianza alto

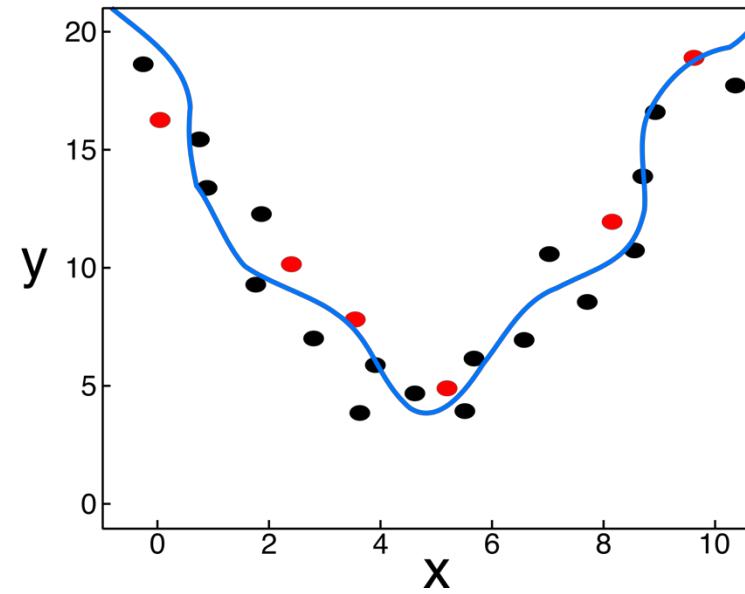
# ESTRATEGIAS

## Regularización

Sin regularizar



Con regularización

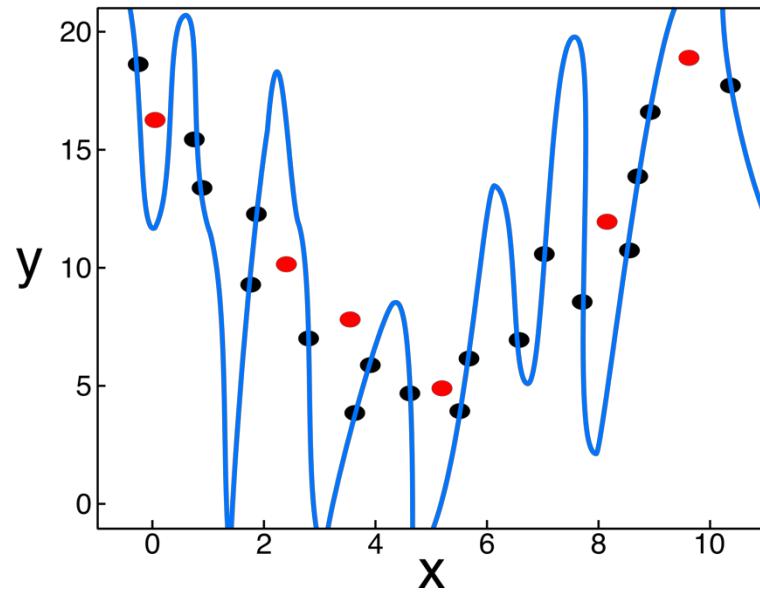


Error de varianza alto

# ESTRATEGIAS

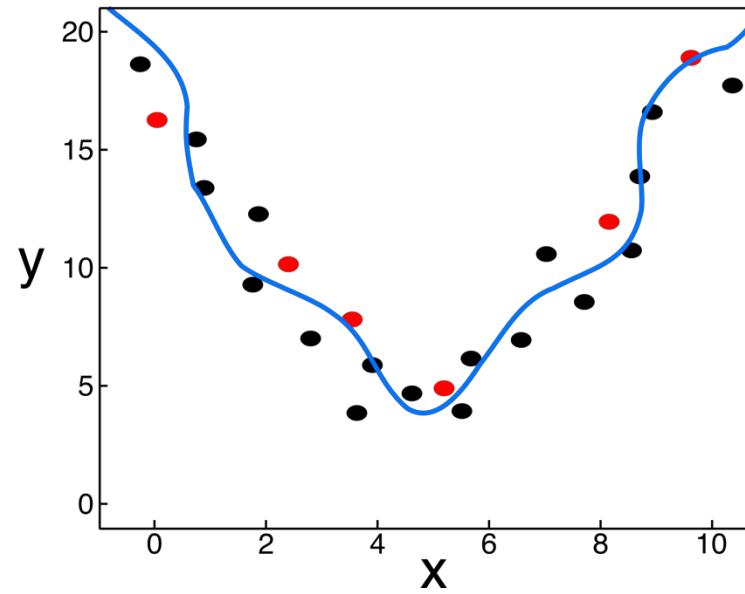
## Regularización

Sin regularizar

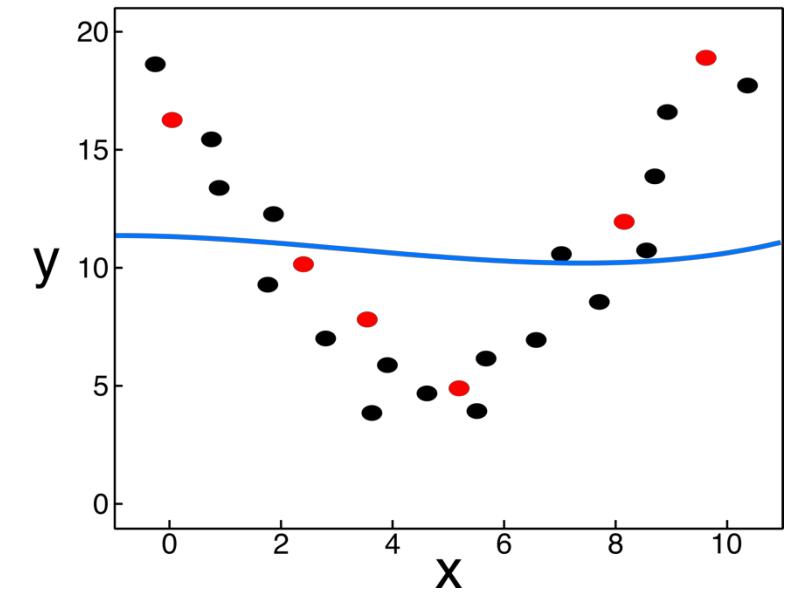


Error de varianza alto

Con regularización



Mucha regularización



Error de sesgo alto