



Aprendizaje Automático

Tarea, Experiencia, Performance

Laura de la Fuente, Hernán Bocaccio

Ayudantes: Gastón Bujía, Diego Onna y Sofía Morena del Pozo

Dirección de e-mail de la materia:

datawillconfess@gmail.com

Aprendizaje Automático:

Samuel (1959)

Campo de estudio que le da a las computadoras la habilidad de aprender sin ser programadas de manera explícita.

Mitchell (1998)

Un programa de computadora se dice que aprende de una **experiencia E** con respecto a una clase de **tareas T** y una medida de **performance P**, si su performance en las tareas T, medidas por P, mejoran con la experiencia E.

Aprendizaje Automático:

Un programa **aprende** una tarea si su performance mejora con la experiencia, y el aprendizaje es **automático** porque no le enseñamos a realizar esa tarea de manera explícita.

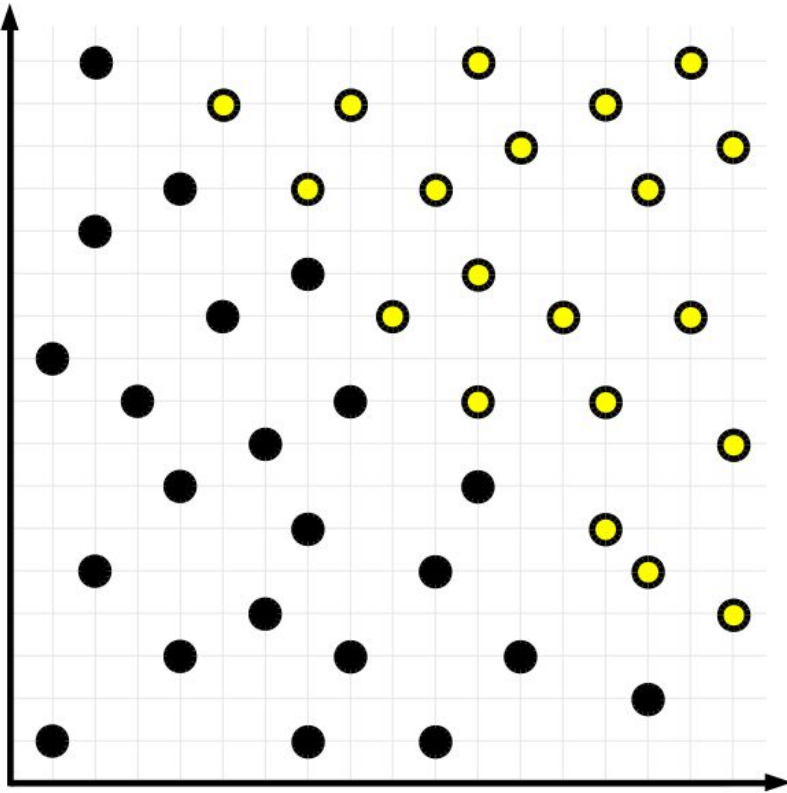
Tenemos que definir

Tarea: objetivo

Experiencia: datos, ejemplos

Medida de performance: distancia formal a mi objetivo

Ejemplos



Tarea:

Predecir el color de un punto

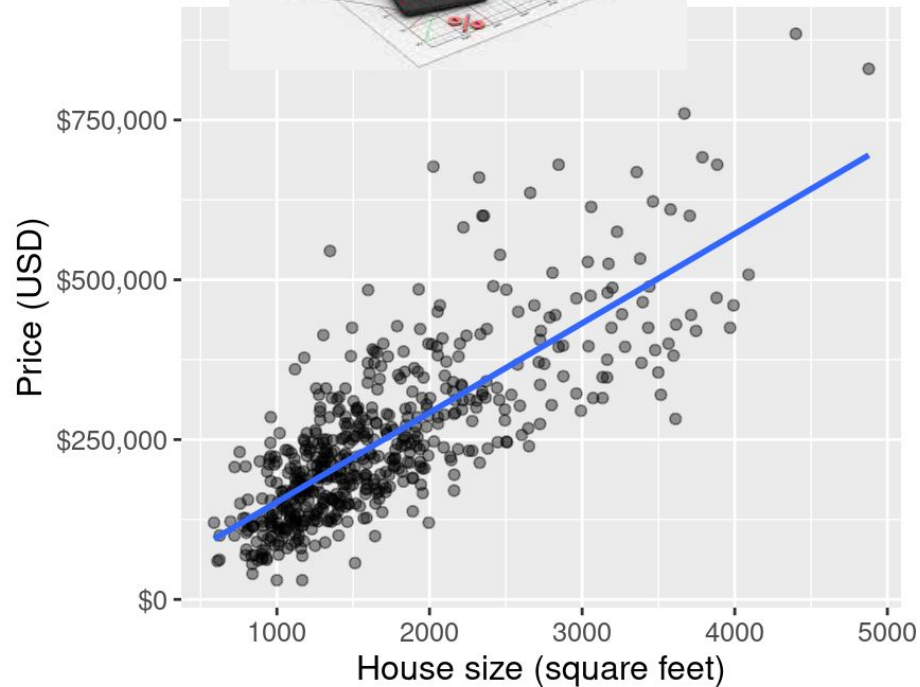
Experiencia:

Base de datos de puntos con su color correspondiente

Medida de performance:

% puntos coloreados correctamente

Ejemplos



Tarea:

Predecir el valor de una casa según su tamaño

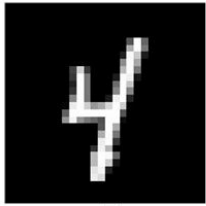
Experiencia:

Datos de casas con su valor y tamaño

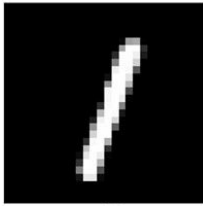
Medida de performance:

Error cometido en el precio predicho

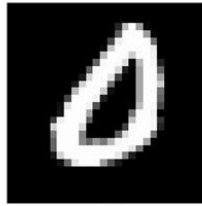
Ejemplos



4 (4)



1 (1)



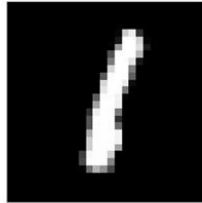
0 (0)



7 (7)



8 (8)



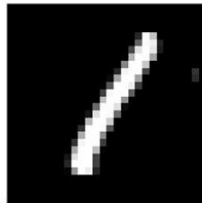
1 (1)



2 (2)



7 (7)



1 (1)

Tarea:

Digitalizar números escritos a mano

Experiencia:

Imágenes de números a mano con sus etiquetas

Medida de performance:

% de transcripciones correctas

Ejemplos



Tarea:

Discriminar si es un chihuahua o un muffin

Experiencia:

Fotos de chihuahuas y muffins con su etiqueta

Medida de performance:

Cantidad de aciertos en la discriminación

Ejemplos

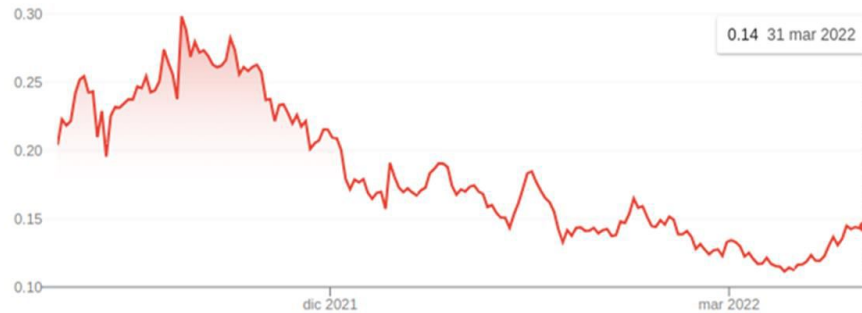
Resumen de mercado > Dogecoin

0.14 USD

-0.06 (29.10 %) ↓ en los últimos 6 meses

31 mar 14:02 UTC · [Renuncia de responsabilidad](#)

1D | 5D | 1 M | **6 M** | UAHF | 1A | 5A | Máx.



1 | DOGE ▼ | 0.14 | USD ▼

Tarea:

Predecir cuánto va a valer una criptomoneda

Experiencia:

Valor de la criptomoneda en el pasado

Medida de performance:

Fracción de predicciones correctas

Ejemplos



Tarea:

Manejo autónomo de vehículos usando sensores

Experiencia:

Mediciones y acciones durante conducción humana

Medida de performance:

Distancia promedio recorrida hasta cometer un error

Otros ejemplos

Sistemas de recomendación



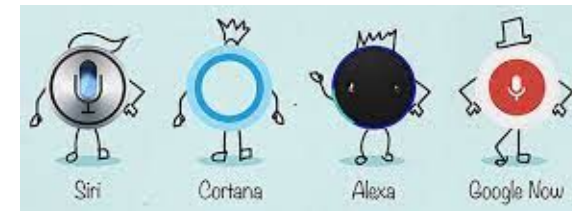
Traductores



Predicción de tiempo de viaje, camino óptimo



Reconocimiento del habla, asistentes virtuales



Publicidad online, chatbots, text2image, detectar spam, reconocimiento de rostros/patentes, diagnósticos clínicos, vehículos autónomos, jugar al Go, ...

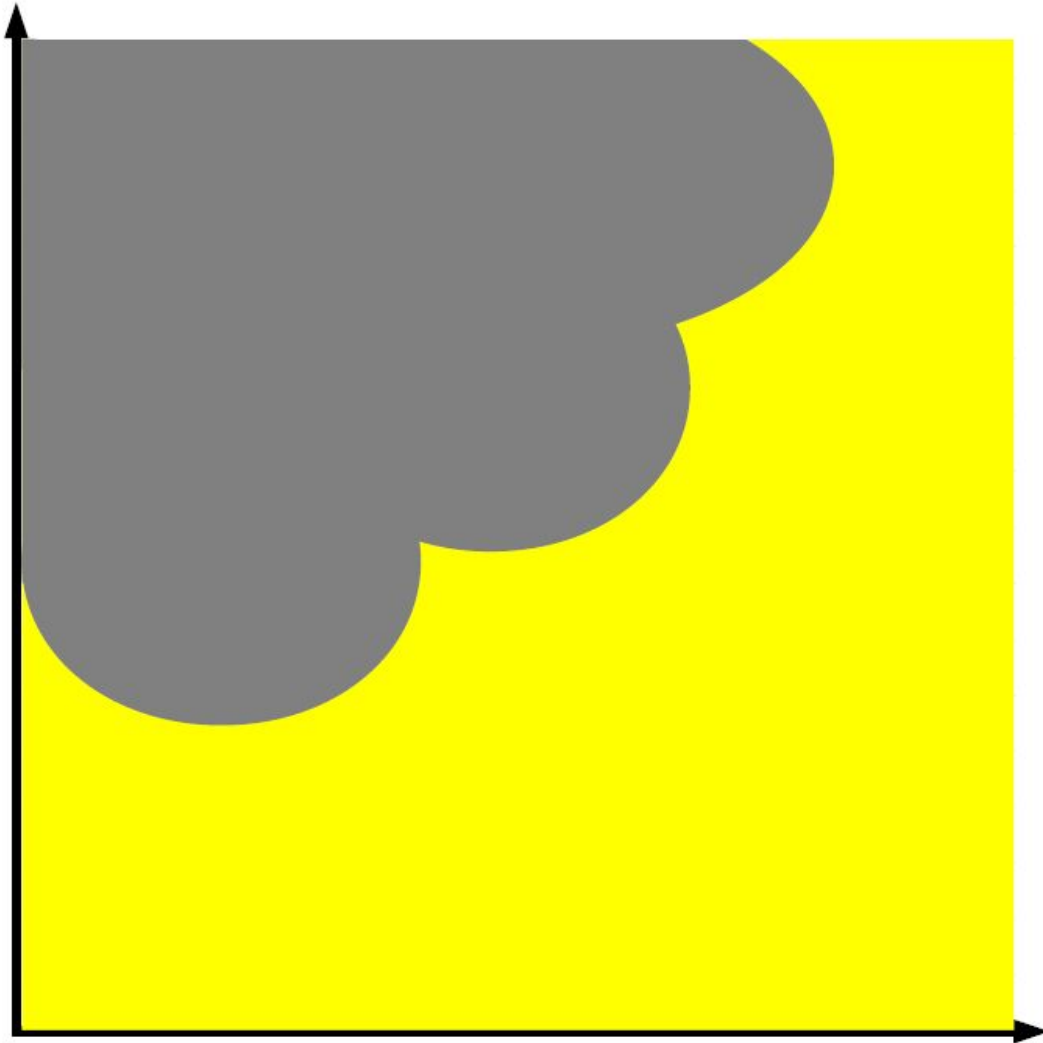
Aprendizaje de conceptos

Concepto: Representación mental asociada a un significante lingüístico.

¿Qué animales son “aves”?



Aprendizaje de conceptos



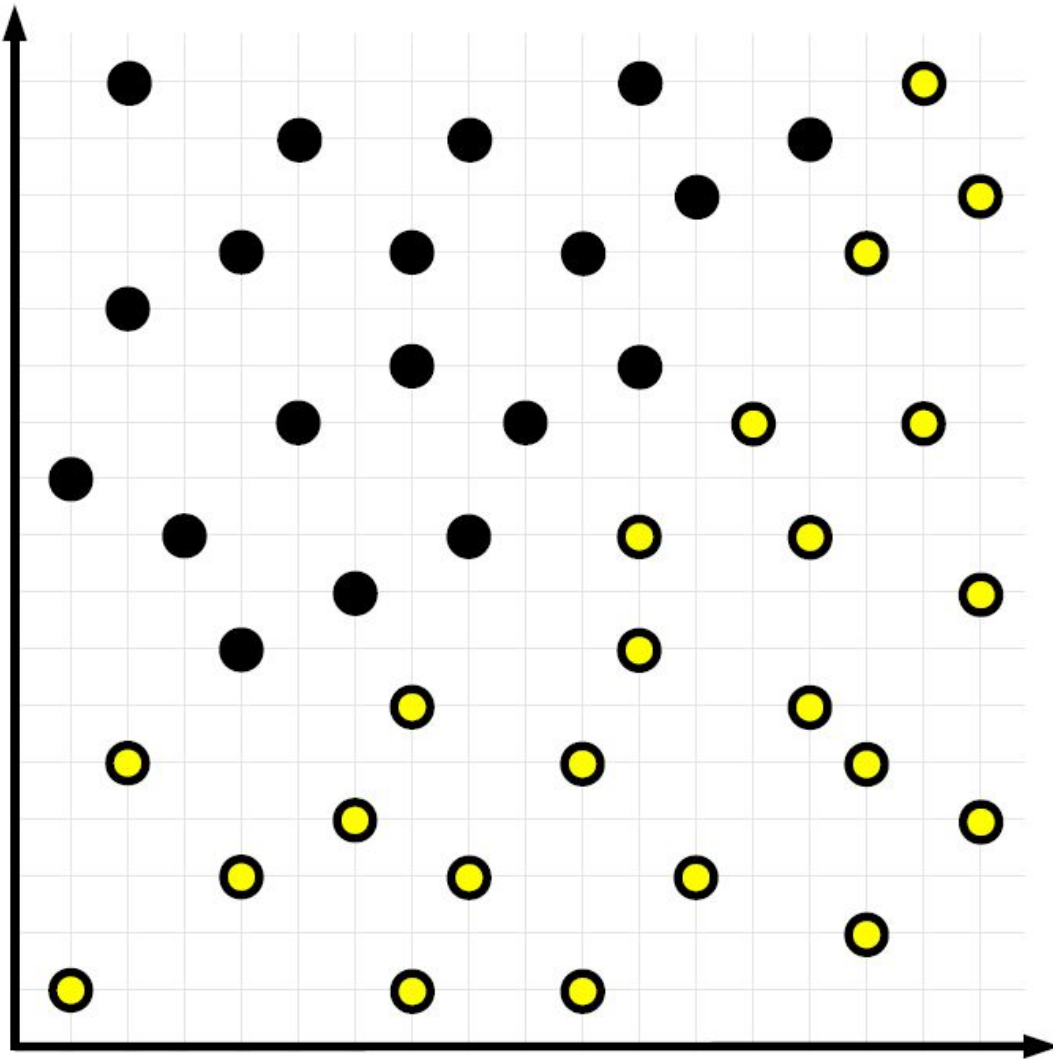
Este es el concepto [función objetivo]
que queremos aprender:

$f(x, y) \rightarrow \text{color}$

¡Es desconocida!

Sólo podemos conocerlo a través de
muestras, inspeccionando datos.

Aprendizaje de conceptos



Tenemos N muestras [instancias].

Cada punto tiene 2 coordenadas: (x, y) [atributos].

Cada punto tiene un color: negro o amarillo [clases].

Buscamos una función

$h(x, y) \rightarrow \text{color}$ [hipótesis o modelo] que aproxime a la función objetivo.

Aprendizaje Inductivo

Consiste en construir un modelo general a partir de información específica.

Principio de Aprendizaje Inductivo

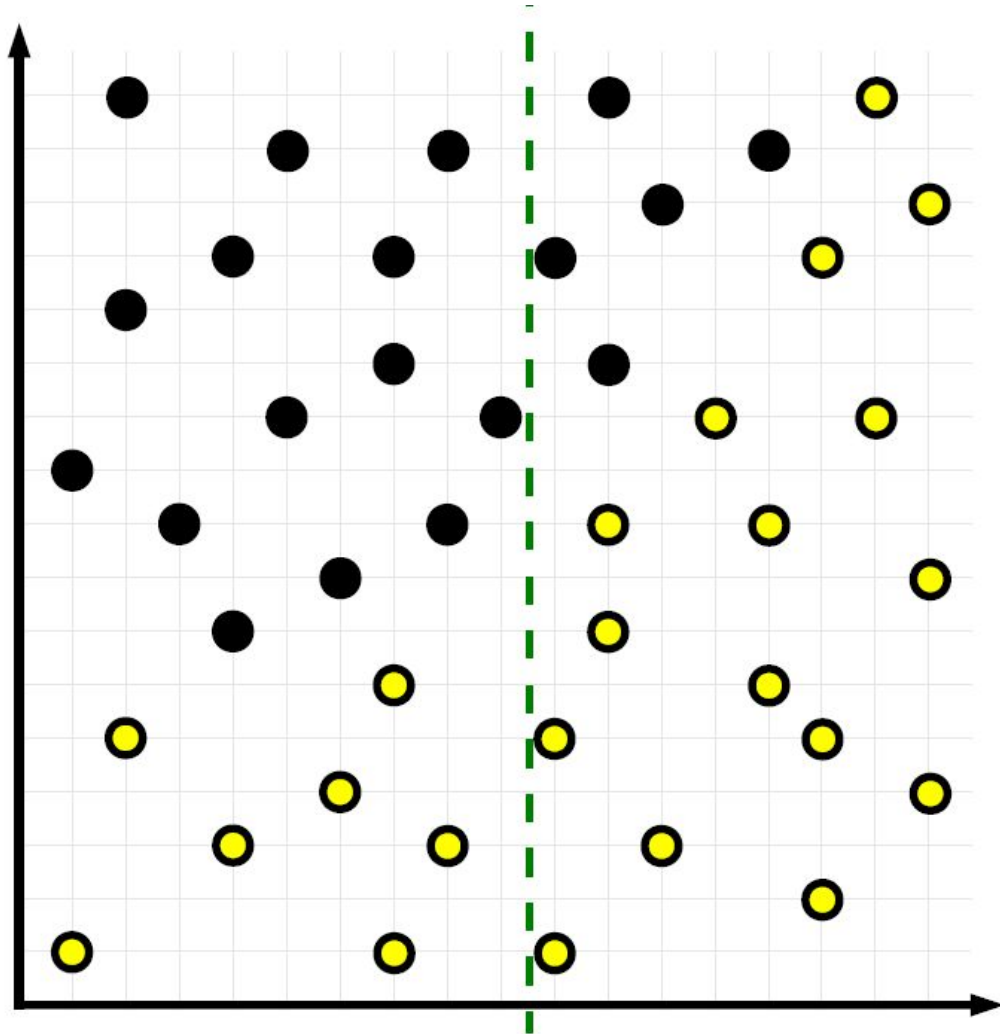
Cualquier **hipótesis h (modelo)**, perteneciente a un **espacio de hipótesis H** que determina la forma que pueda tener, que aproxime bien a una **función objetivo f** sobre un **conjunto suficientemente grande de datos** también aproximará bien a la función objetivo sobre datos no observados (**generaliza**).

La **forma de las hipótesis** (número y tipo de parámetros); y las características de funcionamiento del algoritmo (**cómo recorre el espacio de hipótesis** hasta elegir un único modelo), son parte de las elecciones que uno hace al modelar y contribuyen al **sesgo inductivo**.

Aprendizaje Inductivo

Buscamos:

Función $f(x, y) \rightarrow \text{color}$



Primera idea:

$h(x, y) = \text{colorizq si } x < k$

el otro color, si no

$H = \text{posibles valores de } \langle k, \text{colorizq} \rangle$

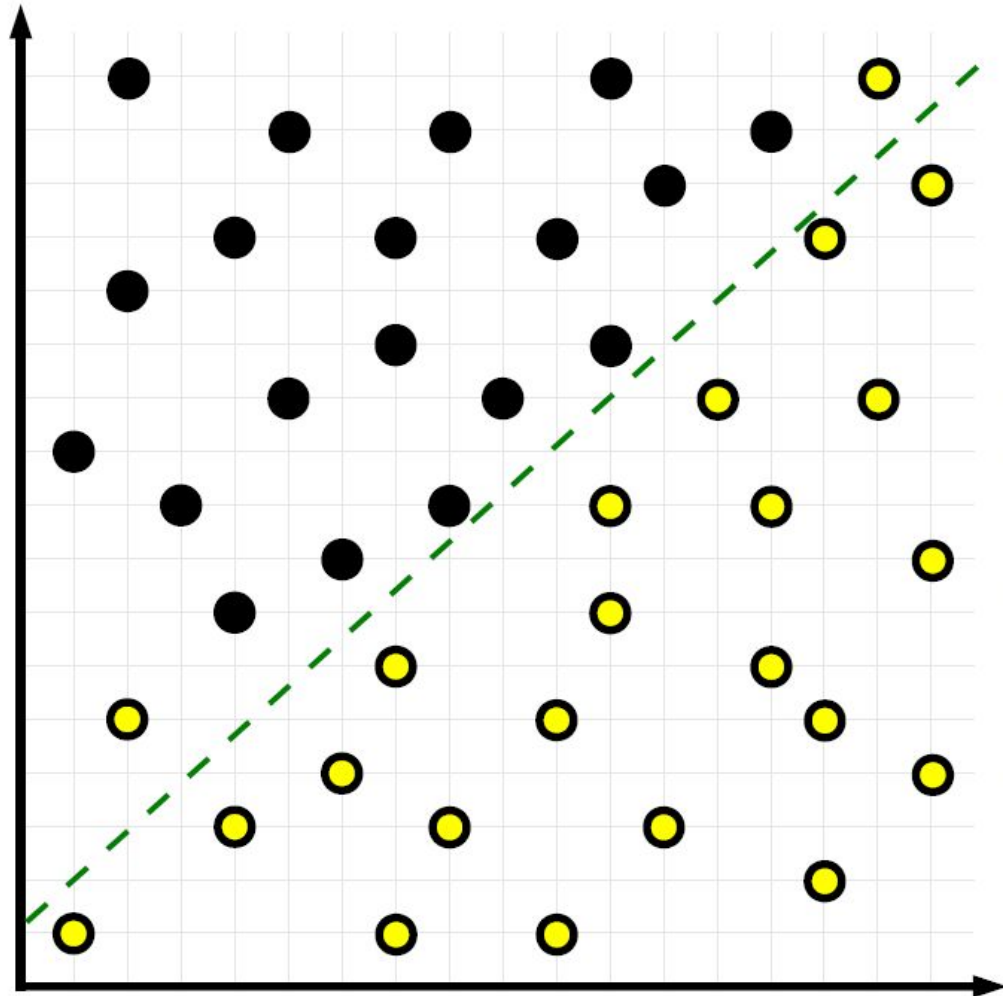
Algoritmo: buscar $\langle k, \text{colorizq} \rangle$ que mejor separe las 2 clases.

Obs: El concepto objetivo no tiene esta forma.

Aprendizaje Inductivo

Buscamos:

Función $f(x, y) \rightarrow \text{color}$



Segunda idea:

$h(x, y) = \text{color}_{\text{sup}}$ si $y > m x + b$

el otro color, si no

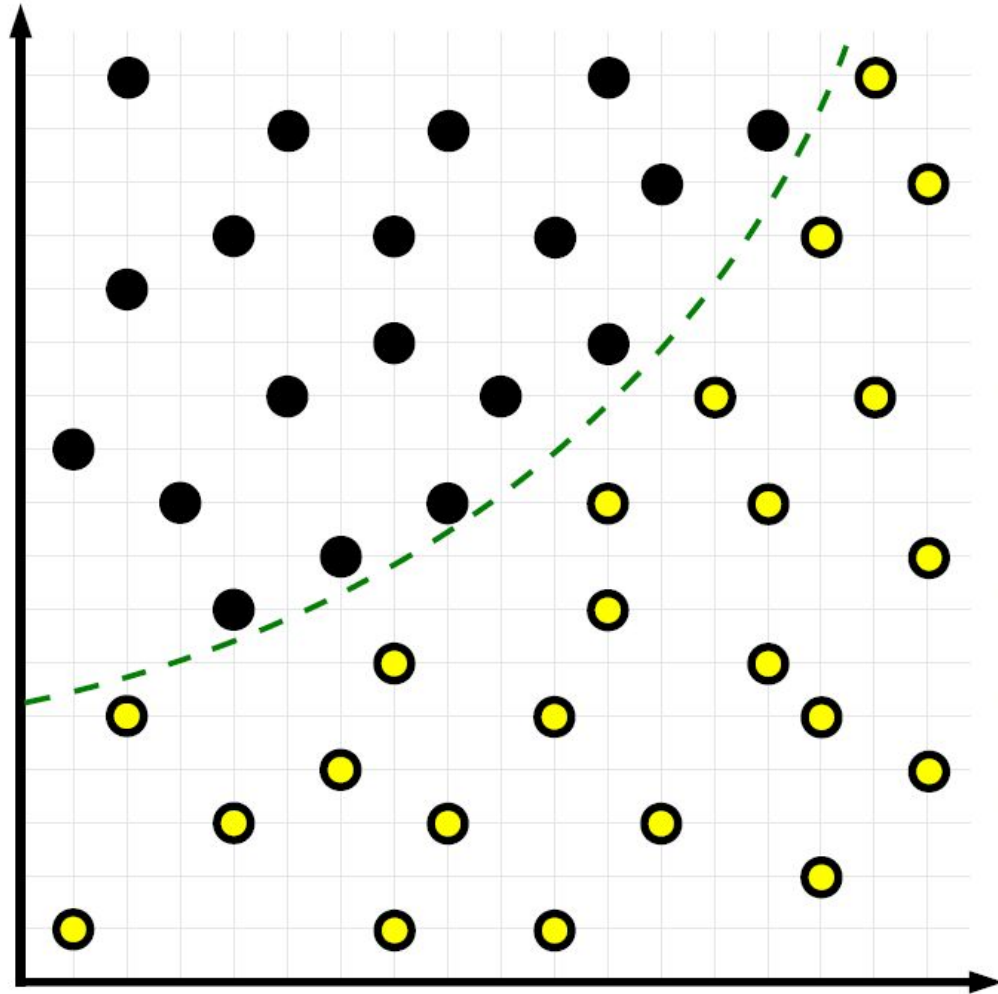
$H = \langle m, b, \text{color}_{\text{sup}} \rangle$

Algoritmo: buscar $\langle m, b, \text{color}_{\text{sup}} \rangle$ que mejor separe las 2 clases.

El concepto objetivo tampoco tiene esta forma, pero la aproximación parece mejor que la anterior.

Encontrar los valores $m, b, \text{color}_{\text{sup}}$ es más costoso.

Aprendizaje Inductivo



Buscamos:

Función $f(x, y) \rightarrow \text{color}$

Tercera idea:

$h(x, y) = \text{color}_{\text{sup}}$ si $y > a x^2 + b x + c$

el otro color, si no

$H = \langle a, b, c, \text{color}_{\text{sup}} \rangle$

Algoritmo: buscar $\langle a, b, c, \text{color}_{\text{sup}} \rangle$ que mejor separe las 2 clases.

El concepto objetivo tampoco tiene esta forma.

La aproximación parece mejor que las otras dos.

Encontrar los valores $a, b, c, \text{color}_{\text{sup}}$ es aún más costoso.

Tipos de modelos según la Tarea

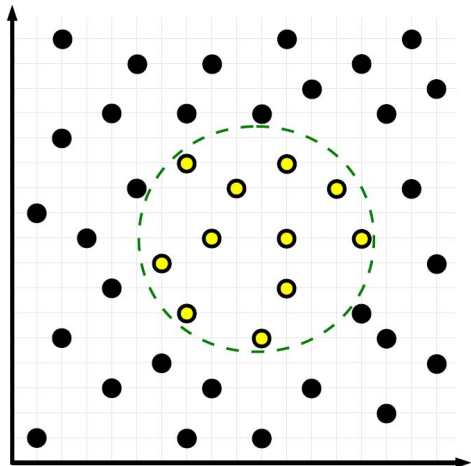
Modelos Predictivos:

Es lo que vamos a ver

Entrenados para realizar la tarea de predecir características en datos nuevos (no vistos en el entrenamiento).

Modelos Generativos:

Entrenados para realizar la tarea de generar datos.



A veces un tipo de modelo podría usarse para las 2 cosas, pero según la tarea puede cambiar cómo lo construyo.

Puedo pedirle al modelo que haga la predicción de color en algún punto nuevo según su posición, o puedo pedirle que me genere la posición de un posible punto amarillo.



Tipos de modelos según la Experiencia

La **experiencia** estará dada por **datos** descritos como **variables**.

Variable: representación de una característica observable en los individuos de una población. Se obtiene mediante diferentes procedimientos de medición.

- Variable cualitativa: sus valores designan una categoría (*sexo / monotributista*)
- Variable cuasi-cuantitativa: sus valores indican un orden o jerarquía (*mucho / poco / nada*)
- Variables cuantitativas: sus valores son cantidades numéricas:
 - Cuantitativa discreta: sus valores son puntos aislados y no tienen un valor consecutivo (*número de hijos*)
 - Cuantitativa continua: pueden tomar cualquier valor dentro de un intervalo numérico (*peso / altura*)

Tipos de modelos según la Experiencia

Aprendizaje supervisado:

Los **datos están anotados** con la respuesta correcta que quiero predecir.

- Clasificación: Predecir una clase (variable categórica)
- Regresión: Predecir un valor numérico (variable numérica)

Aprendizaje no supervisado:

Los **datos** de entrenamiento **no están anotados**.

- Encontrar patrones, clustering.
- Transformar datos, reducción de dimensionalidad.

Aprendizaje por refuerzos:

Los datos surgen de la **interacción con el entorno**, y el aprendizaje es gradual en base a una **recompensa**.

Aprendizaje supervisado

Los **datos están anotados** con la respuesta correcta que quiero predecir.

The diagram shows a table with 5 columns and 15 rows. The first four columns are grouped under the label 'atributos' (attributes) with a red bracket. The fifth column is labeled 'clase' (class) with a red arrow pointing to its header. A red bracket on the left side of the table is labeled 'instancias' (instances). The table data is as follows:

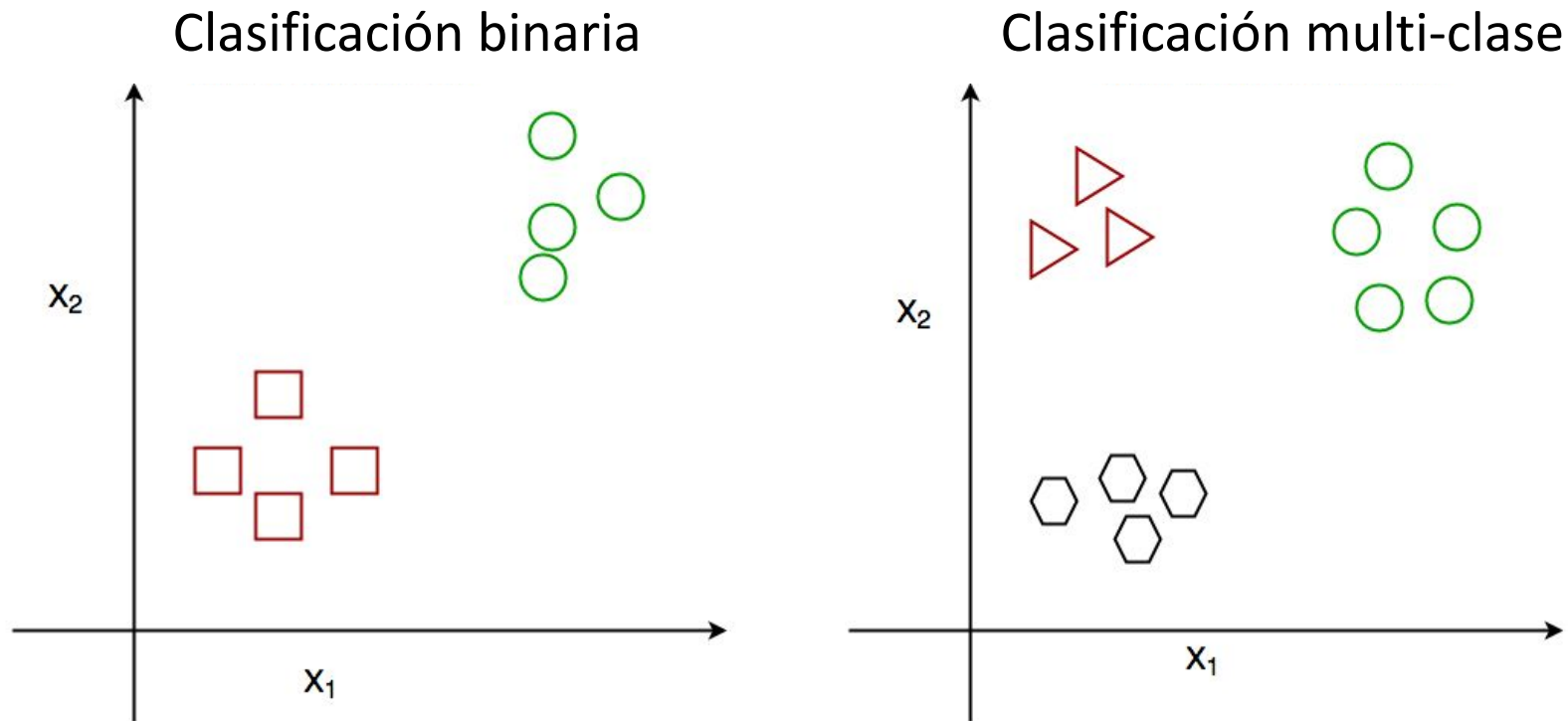
atributos				clase
Cielo	Temperatura	Humedad	Viento	¿Camina?
Sol	Calor	Alta	Débil	No
Sol	Calor	Alta	Fuerte	No
Nublado	Calor	Alta	Débil	Sí
Lluvia	Templado	Alta	Débil	Sí
Lluvia	Frío	Normal	Débil	Sí
Lluvia	Frío	Normal	Fuerte	No
Nublado	Frío	Normal	Fuerte	Sí
Sol	Templado	Alta	Débil	No
Sol	Frío	Normal	Débil	Sí
Lluvia	Templado	Normal	Débil	Sí
Sol	Templado	Normal	Fuerte	Sí
Nublado	Templado	Alta	Fuerte	Sí
Nublado	Calor	Normal	Débil	Sí
Lluvia	Templado	Alta	Fuerte	No

En este ejemplo, la **respuesta** correcta que quiero predecir es una variable categórica (clase). Pero también podría ser una variable numérica (metros que camina).

Conseguir datos anotados puede ser **costoso** (tiempo, recursos, dinero), o en el caso de datos minados o de usuarios puede tener problemas de **calidad y sesgos**. También puede haber problemas de disponibilidad o **sensibilidad de la información**.

Clasificación

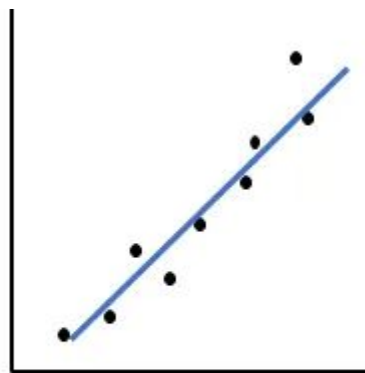
Cuando la **variable** anotada que quiero predecir es **categorica** (clase).



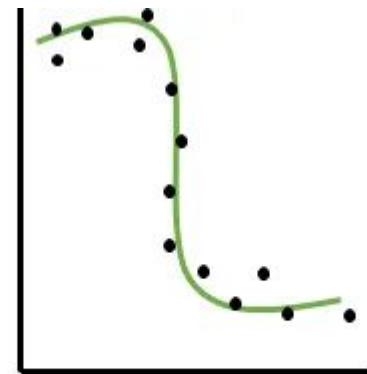
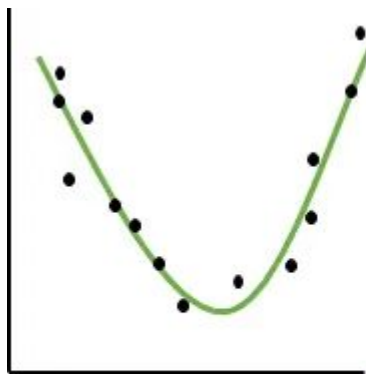
La **predicción** sobre un dato nuevo puede ser la **clase** a la que pertenece o la **probabilidad** de pertenecer a esa clase según el modelo y el algoritmo.

Regresión

Cuando la **variable** anotada que quiero predecir es **numérica** (cantidad).



Regresión lineal



Regresión no lineal (polinomio, etc)

Respuesta

Parámetros

Atributos

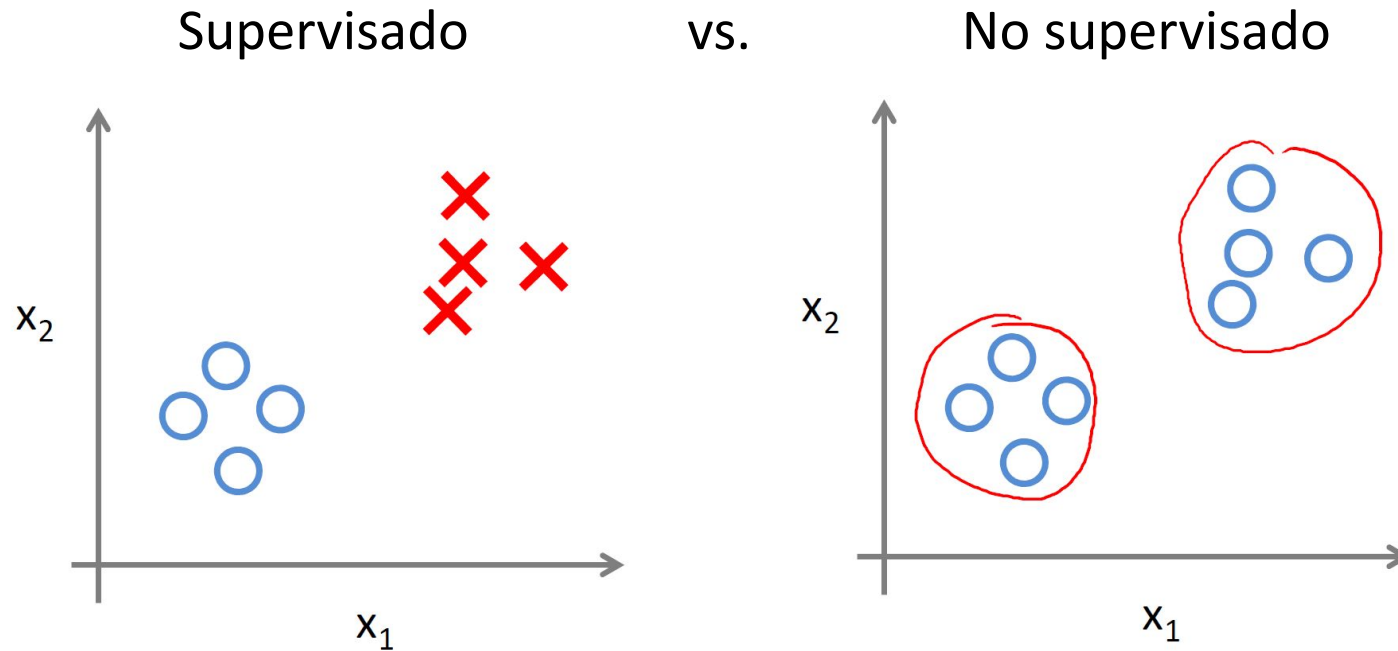
$$Y_i = \beta_0 + \beta_1 X_i$$

Postulo que la **función objetivo** responde a una **relación lineal**.

Exploro el **espacio de hipótesis** definido por rectas genéricas (**parámetros desconocidos**).
Estimo los **parámetros** que aproximan a la función objetivo, y definirán mi **hipótesis**.

Aprendizaje no supervisado

Los **datos** de entrenamiento **no están anotados**.



Se busca encontrar patrones ocultos en los datos o transformarlos para entenderlos mejor.

Clustering, reducción dimensionalidad.
Ejemplos: segmentación de mercado,
análisis redes sociales, ...

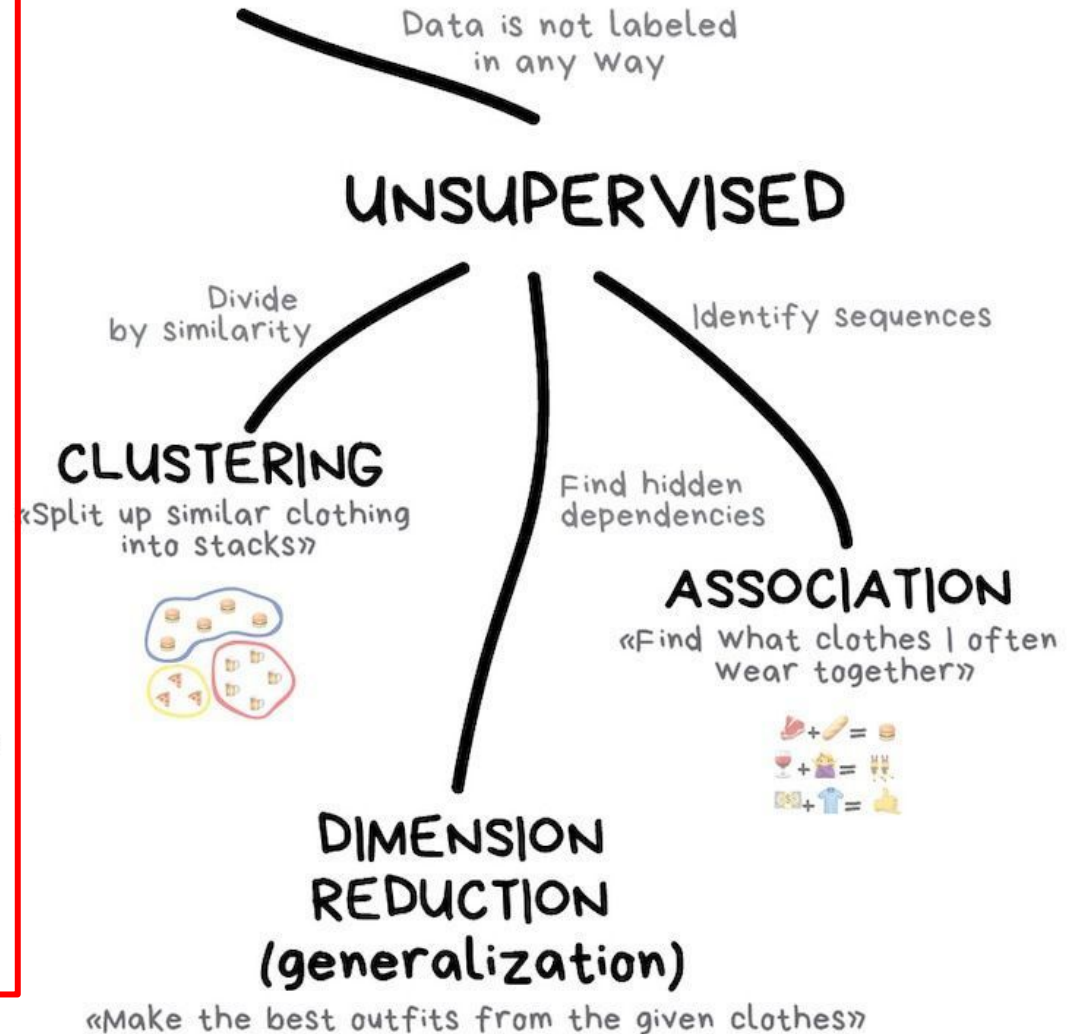
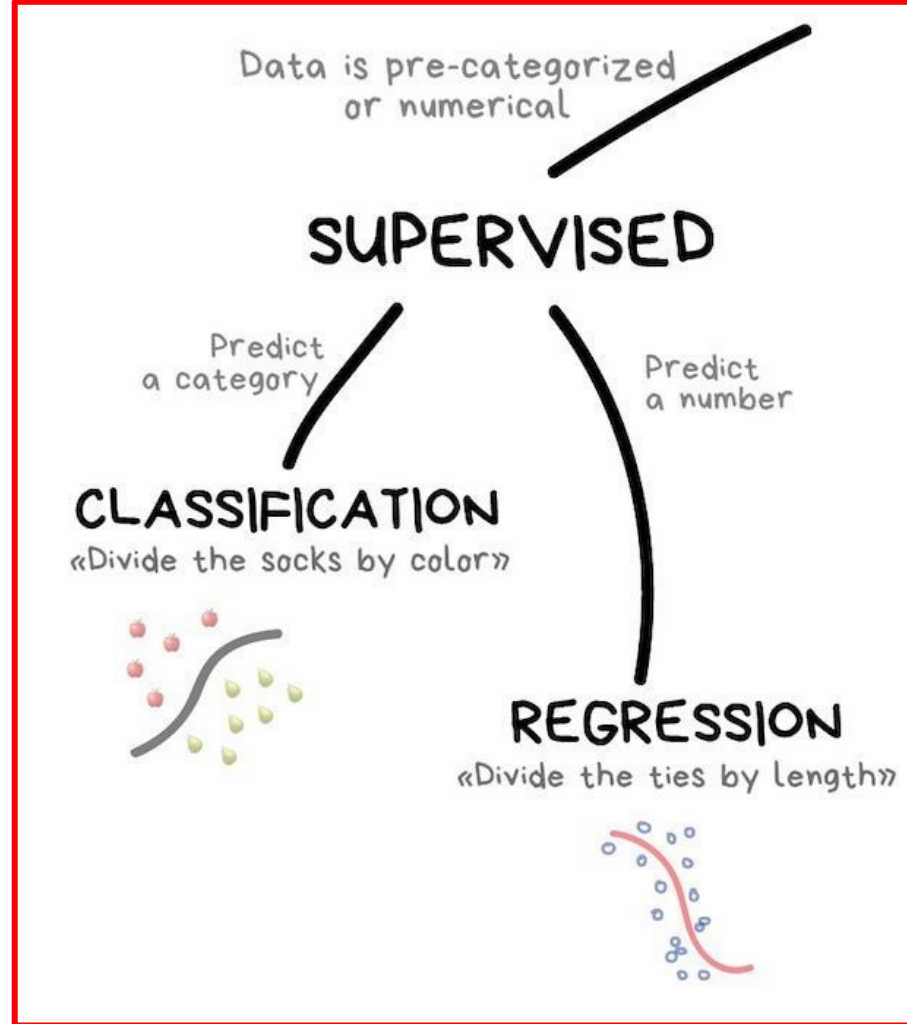


Aprendizaje por refuerzo

- Aprendizaje de un agente autónomo (con sensores) para elegir acciones óptimas que le permitan lograr sus objetivos.
- Premio y castigos, en función de la satisfacción del objetivo, con efecto atrasado.
- Agente tiene que aprender qué secuencias de acciones realizar para obtener el mejor premio posible.
- Ej: juegos de mesa, robots autónomos, cerebro (secuencia de acciones).



CLASSICAL MACHINE LEARNING



Es lo que vamos a ver

Medidas de performance

¿Qué medidas usamos?

Accuracy: proporción de aciertos totales

Ventaja, intuitiva. Desventaja, mala en datos desbalanceados (ej: meteorito)



No cae hasta que es tarde...



Medidas de performance

¿Qué medidas usamos?

Sensibilidad/recall/TVP:

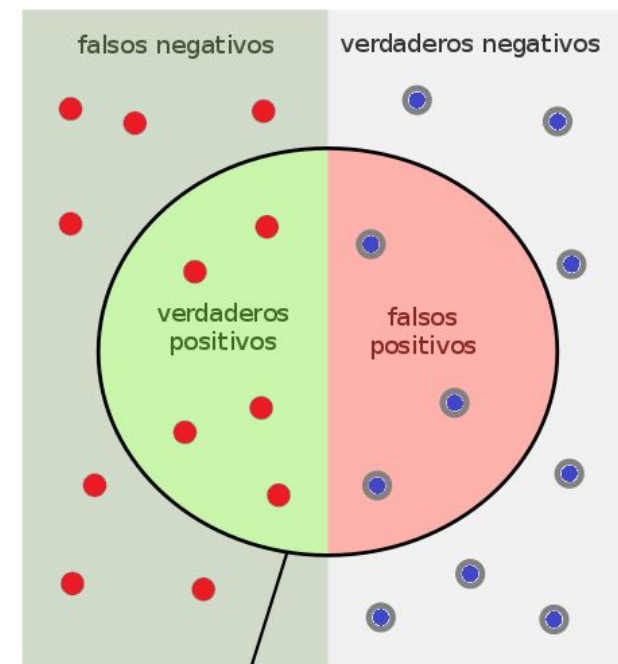
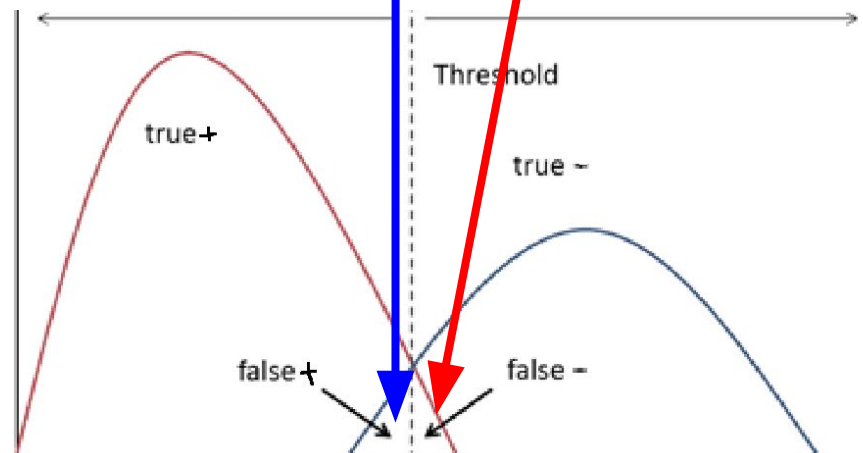
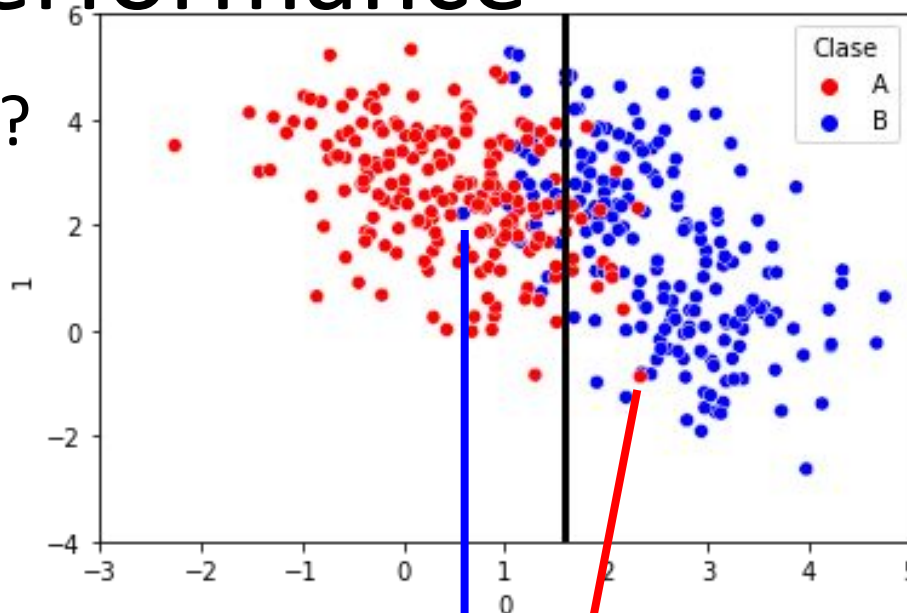
Proporción de positivos que recuperé

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

Especificidad/TVN:

Proporción de negativos que recuperé

$$\text{Especificidad} = \frac{VN}{VN + FP}$$



elementos seleccionados

¿Cuántos objetos relevantes se seleccionaron?
i.e. Cuántas personas enfermas son identificadas como tales.

¿Cuántos elementos negativos se identifican como negativos?
i.e. Cuántas personas sanas son identificadas como no enfermas.

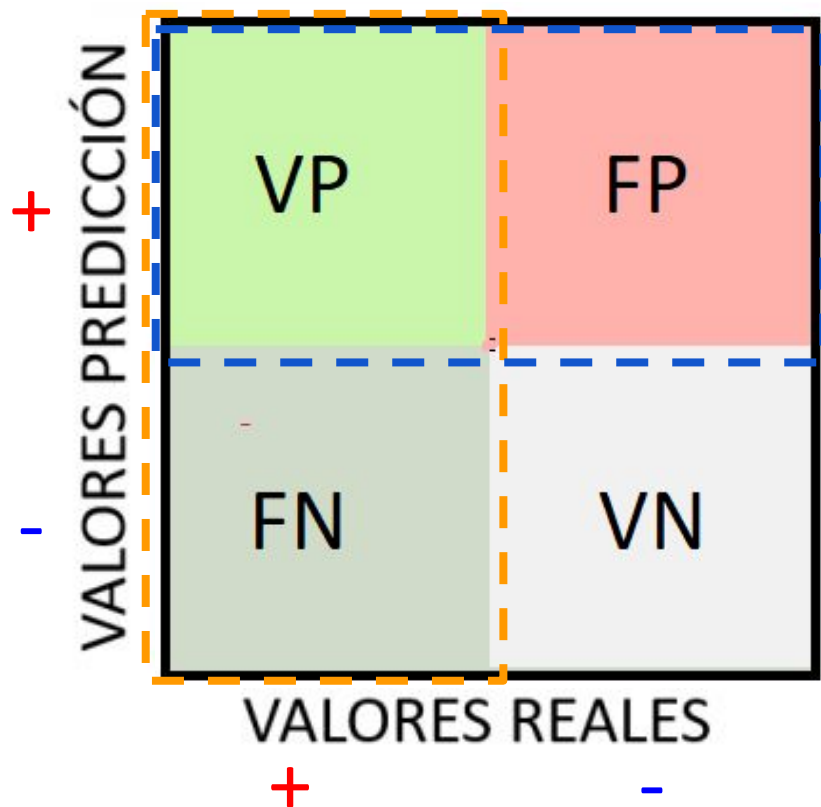
$$\text{Sensibilidad} = \frac{5}{12}$$

$$\text{Especificidad} = \frac{7}{10}$$

Medidas de performance

¿Qué medidas usamos?

Matriz de confusión



$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$



Precision alto: los positivos predichos, lo son (menos error de tipo I)

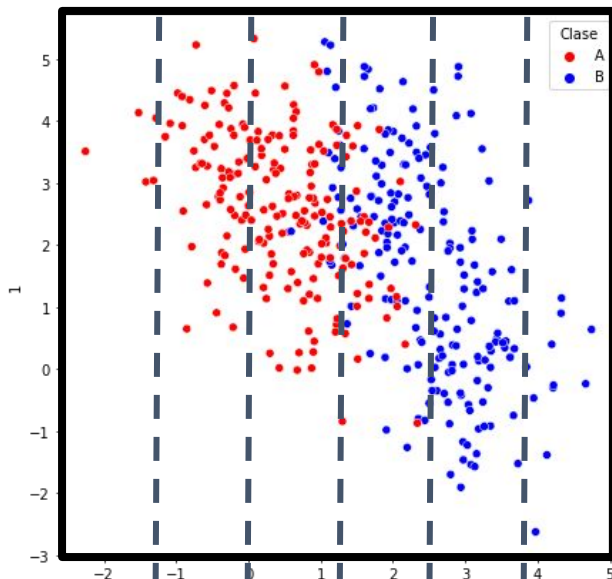


Recall alto: no se me pasan positivos sin predecir (menos error de tipo II)

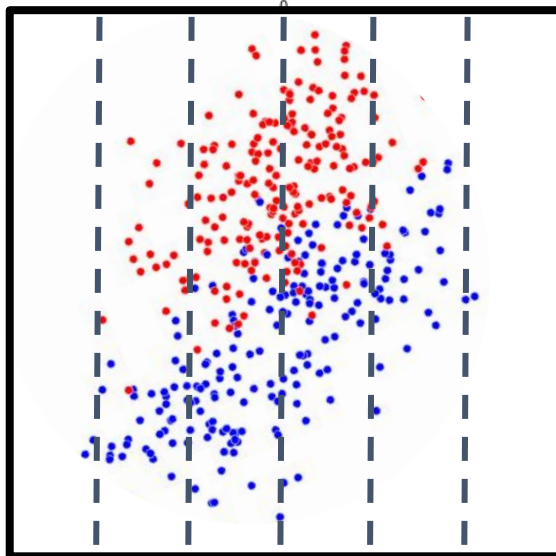


Medidas de performance

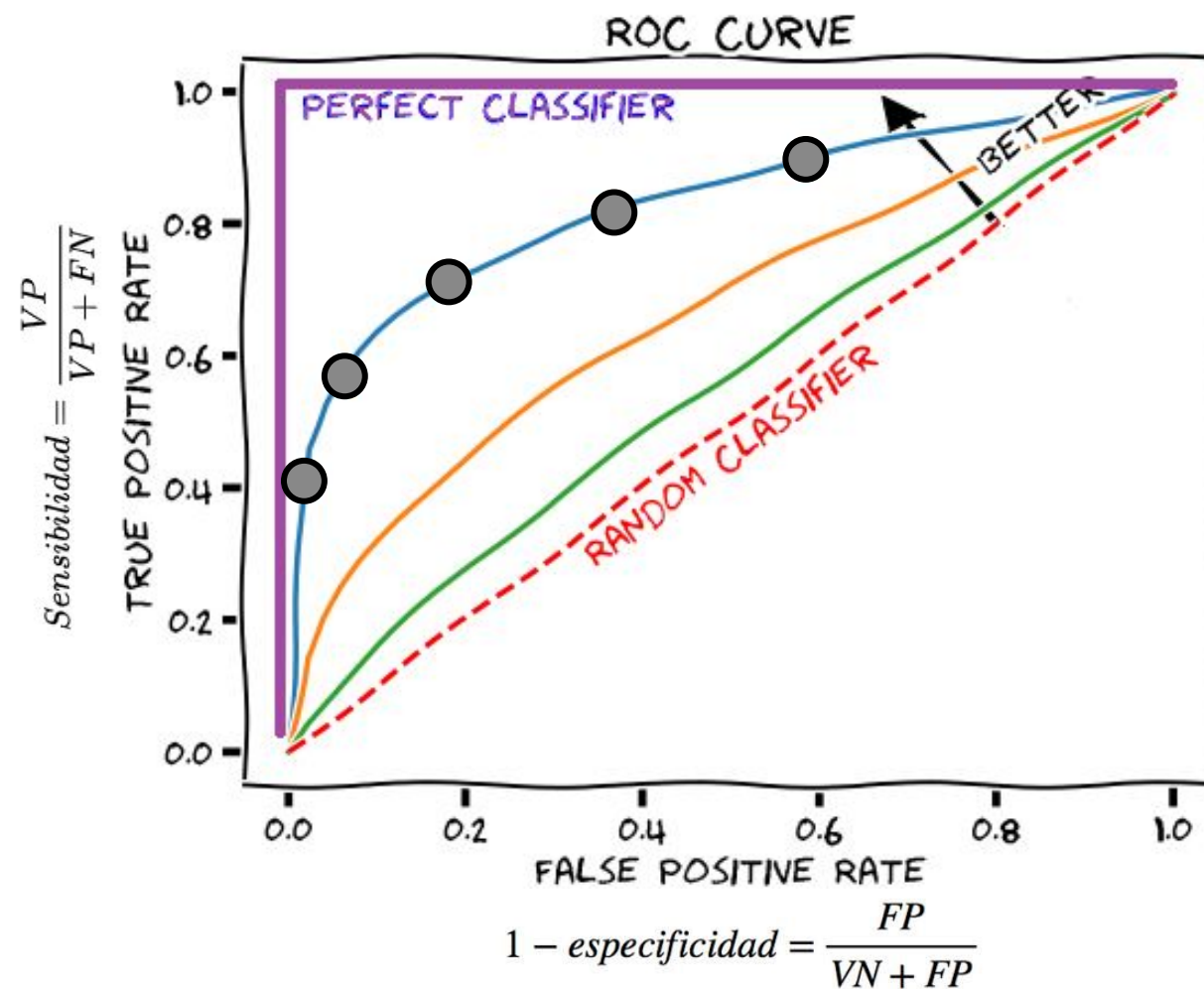
Clases



Random



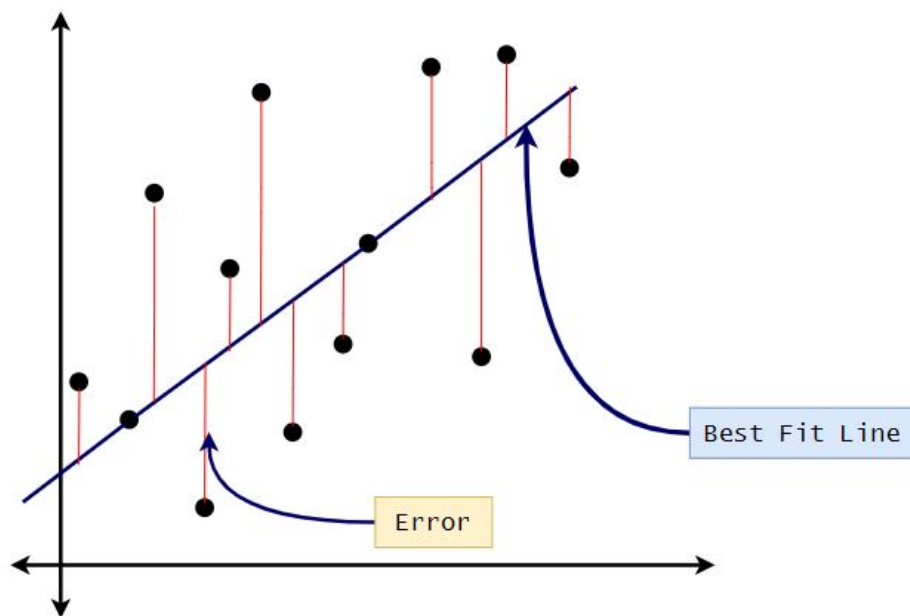
Curvas ROC



Área bajo la curva ROC (AUC) es una medida de desempeño

Medidas de performance

Caso regresión



$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n \underbrace{(y^{(i)} - \hat{y}^{(i)})}_{\text{error}}^2$$

Respuesta



Parámetros

Atributos



$$Y_i = \beta_0 + \beta_1 X_i$$

Hipótesis (estimación)

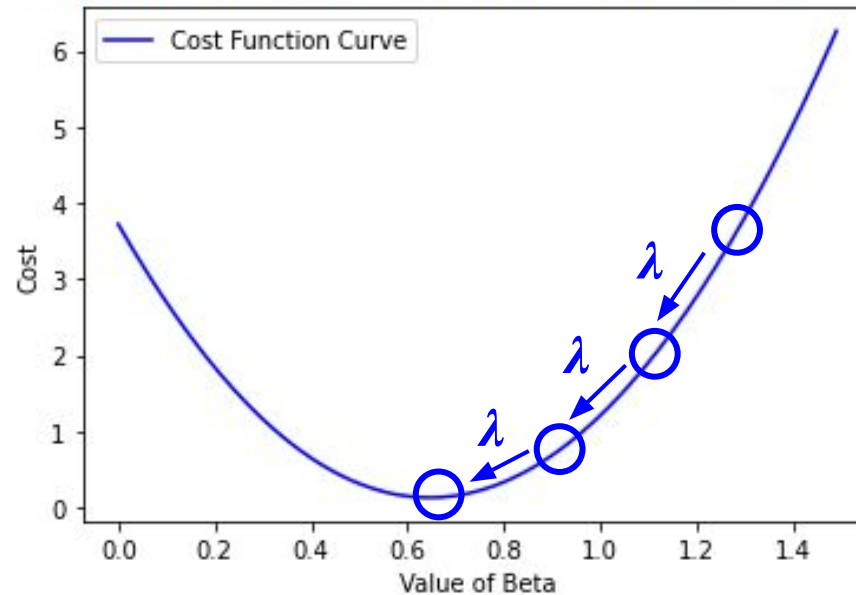
$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$$

Predicción en datos nuevos

Error cuadrático medio

Función costo

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 \cdot x$$



Error cuadrático medio

$$J(\hat{\beta}) = \frac{1}{n} \sum_{i=1}^n \underbrace{(y^{(i)} - \hat{y}^{(i)})}_{\text{error}}^2$$

└─ **Función costo**

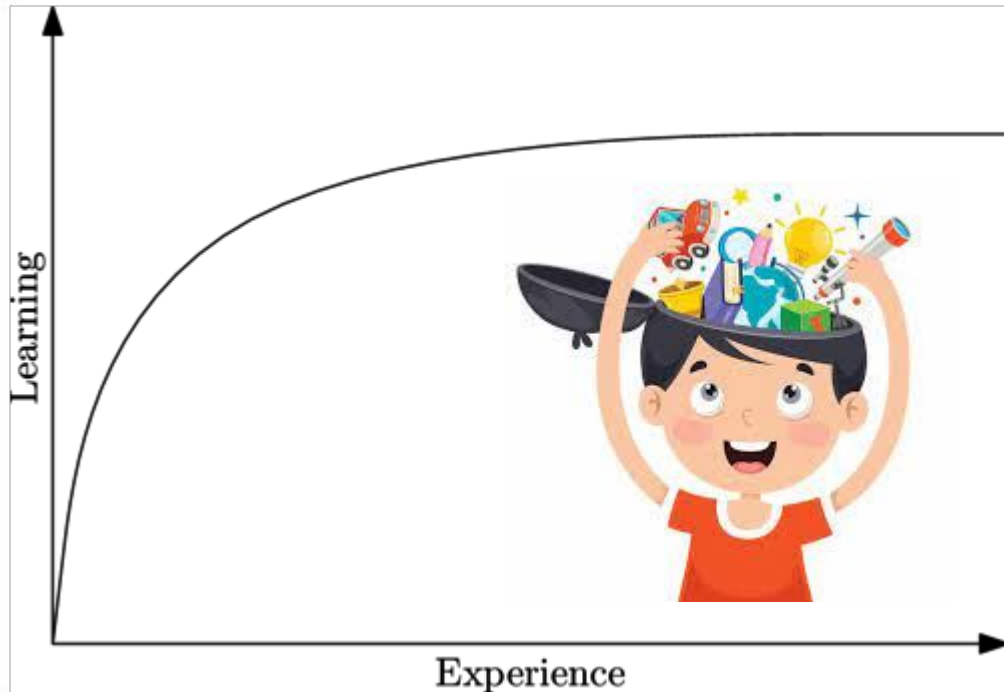
Gradiente: $\frac{\partial J(\hat{\beta})}{\partial \hat{\beta}}$

Entrenar un modelo es básicamente resolver un problema de **optimización** que consiste en encontrar los **parámetros que minimizan una función costo**.

En los problemas de clasificación también se puede definir una función costo a minimizar (depende de modelo y algoritmo).

Uno de los métodos de minimización de la función costo, y el más usado, es el de **descenso por el gradiente**, con un paso llamado **learning rate** (lambda).

Curvas de aprendizaje



De manera general, el **aprendizaje** implica una **mejoría a partir de la experiencia**.

Esta mejoría no es infinita, sino que satura, lo que refleja una curva de aprendizaje.

El caso del aprendizaje automático es análogo. La mejoría se puede cuantificar con la **performance** (que va a maximizarse cuando minimizo la función costo), y la experiencia va a depender de la **cantidad de datos** que usé para el **entrenamiento**.

Cuantos más datos uso, mejora la performance de entrenamiento. Siempre? No, a veces no es suficiente aumentar la cantidad de datos. Puede ser que el modelo no sea el adecuado, que los datos no reflejen el concepto que quiero aprender, o que directamente no haya nada para aprender.

Pasamos a los Colabs...