



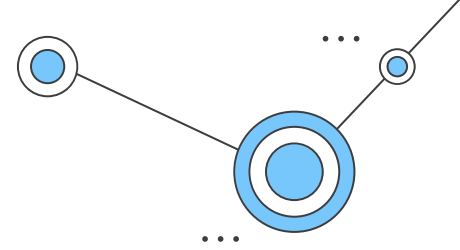
# Introducción al Machine Learning



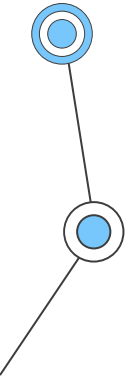
**UNIVERSIDAD  
CATÓLICA**  
DE CÓRDOBA  
JESUITAS

Dr. Francisco Arduh  
2023

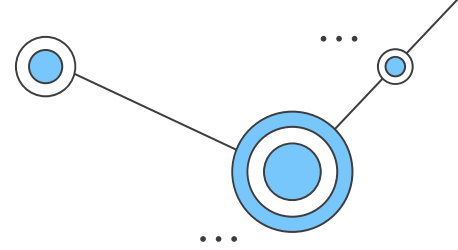
# Definición



“Machine Learning es el campo de estudio que le da a las computadoras la habilidad de aprender sin ser programadas explícitamente” Arthur Samuel, 1959

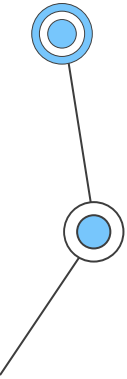


# Definición

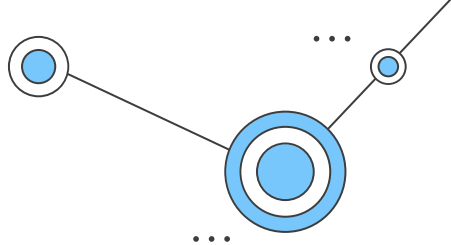


“Machine Learning es el campo de estudio que le da a las computadoras la habilidad de aprender sin ser programadas explícitamente” Arthur Samuel, 1959

“Un programa de computadora se dice que aprende de la experiencia  $E$  con respecto a alguna tarea  $T$  y alguna medida de desempeño  $P$ , si su desempeño en  $T$ , medido por  $P$ , mejora la experiencia  $E$ ” Tom Mitchell, 1997



# ¿Qué es Machine Learning?

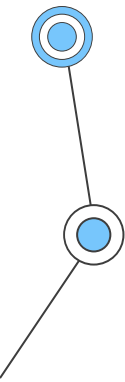


“Machine Learning es el campo de estudio que le da a las computadoras la habilidad de aprender sin ser programadas explícitamente” Arthur Samuel, 1959

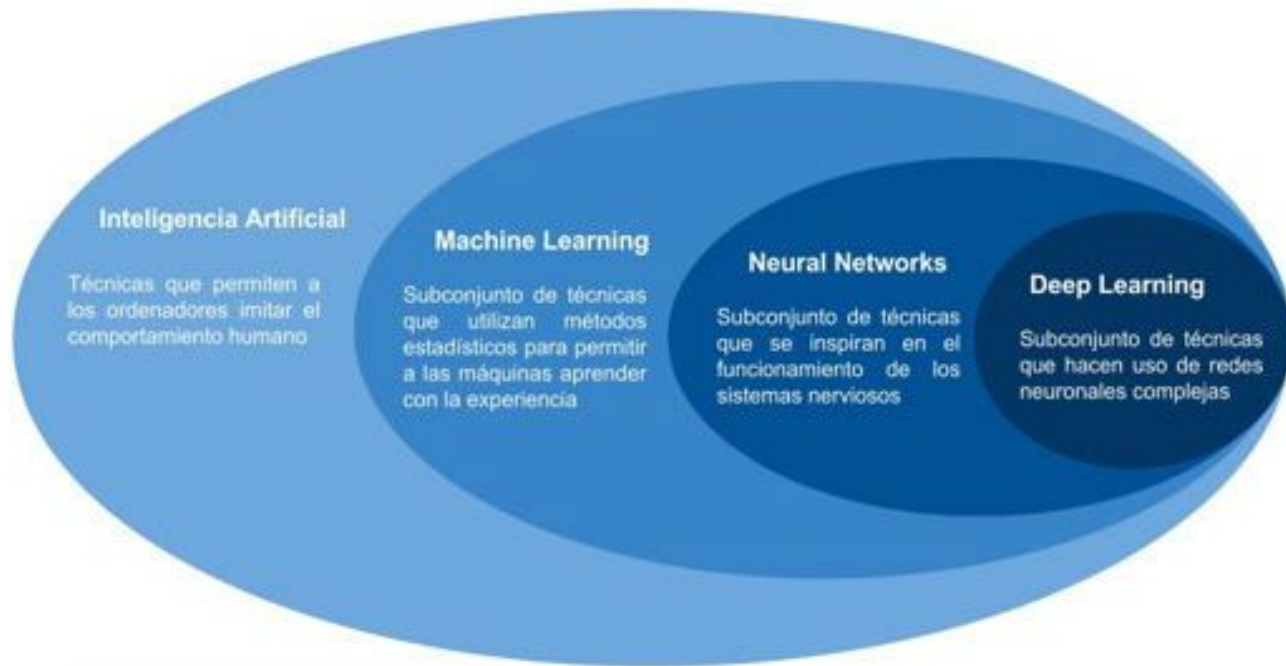
“Un programa de computadora se dice que aprende de la experiencia E con respecto a alguna tarea T y alguna medida de desempeño P, si su desempeño en T, medido por P, mejora la experiencia E” Tom Mitchell, 1997

Nota:

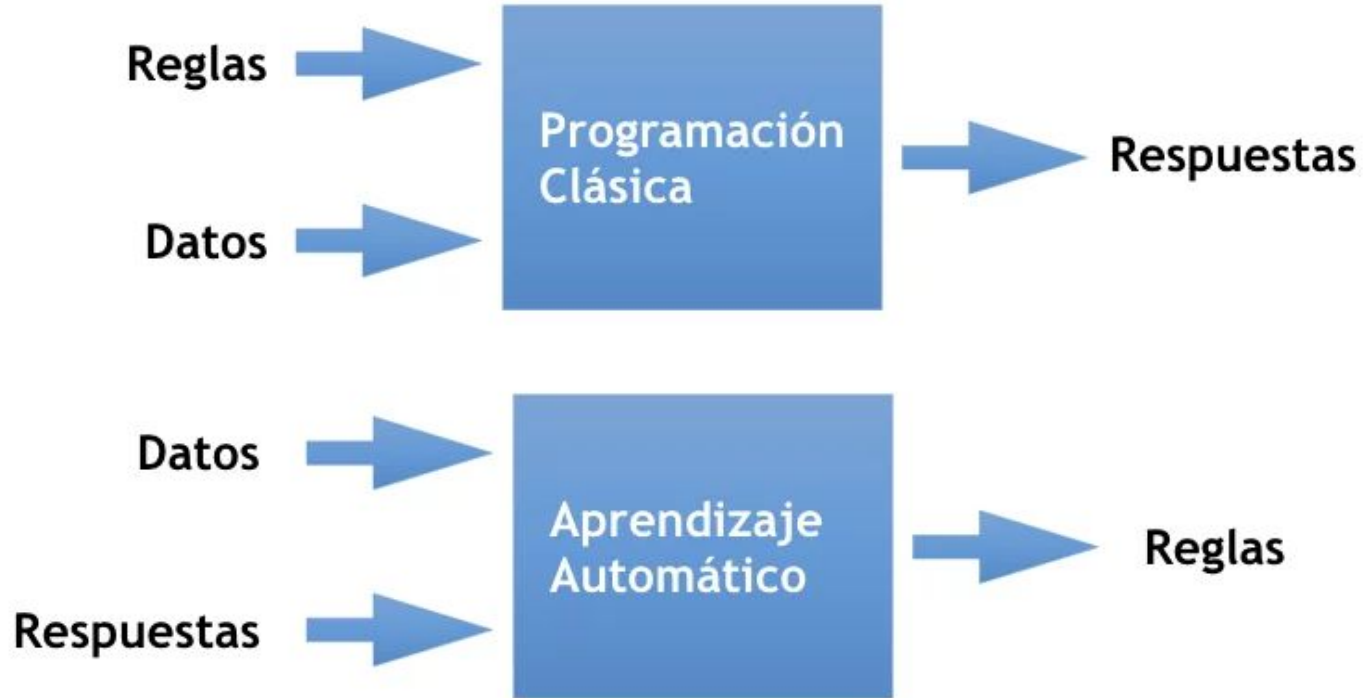
Machine Learning === Aprendizaje automático === Aprendizaje de máquina



# ¿Inteligencia artificial o Machine Learning?

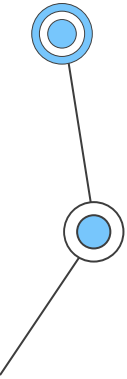
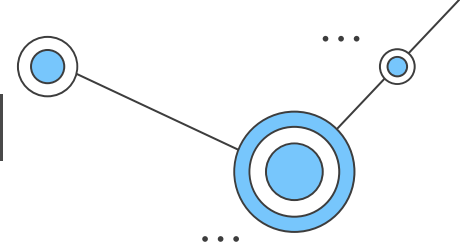


# Diferencias de paradigmas



# Ejemplo: Programación tradicional

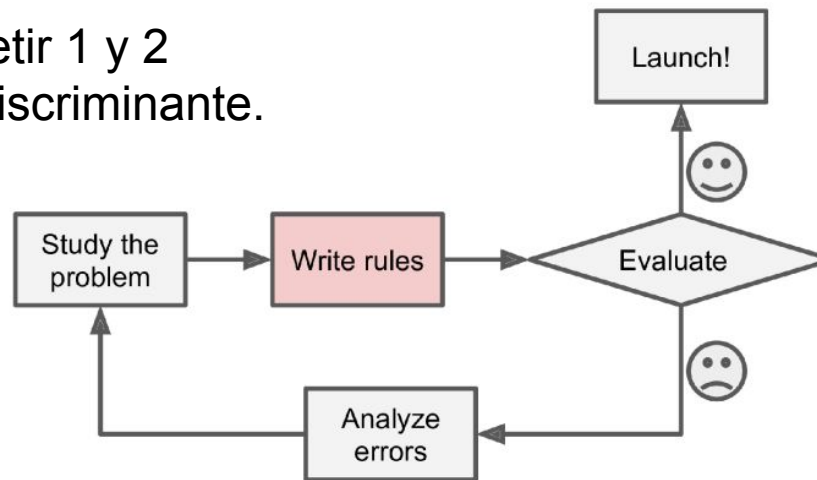
Se quiere separar los email spam de los no spam (ham), una aproximación tradicional seria



# Ejemplo: Programación tradicional

Se quiere separar los email spam de los no spam (ham), una aproximación tradicional seria:

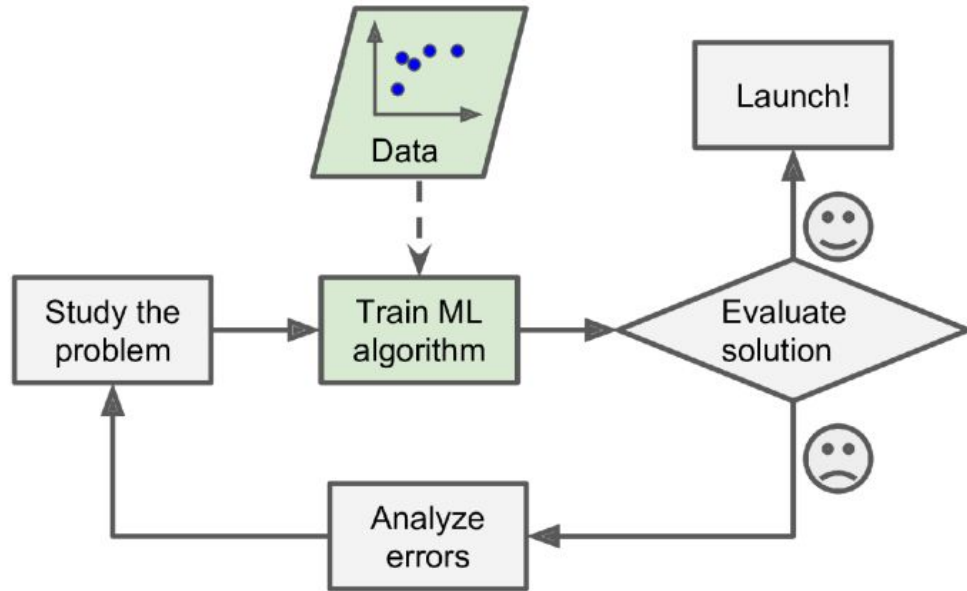
1. Analizar los email y encontrar patrones que nos indican que un email es spam.
2. Escribir un algoritmo con los patrones encontrados.
3. Evaluar el algoritmo anterior y repetir 1 y 2 hasta obtener un buen algoritmo discriminante.





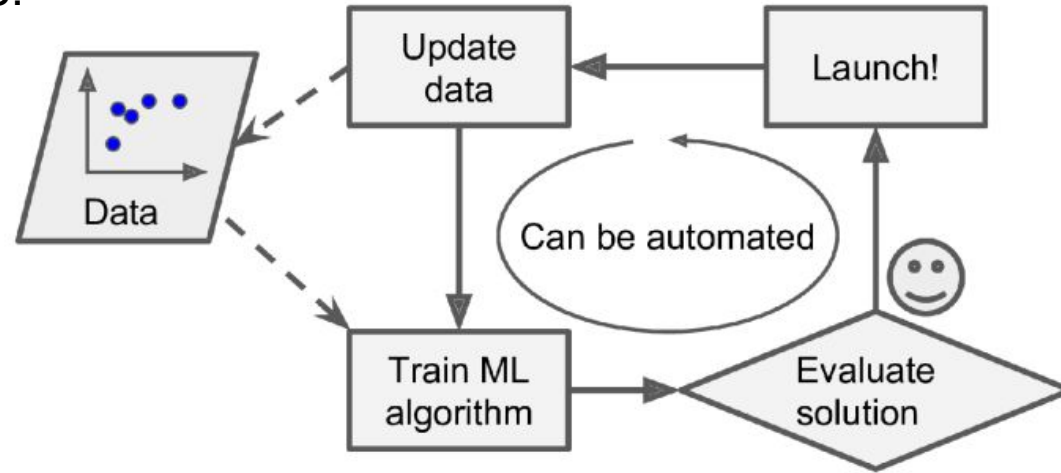
# Ejemplo: Machine Learning

- Aprende qué patrones se repiten más usualmente en spam que en ham para poder distinguirlos.



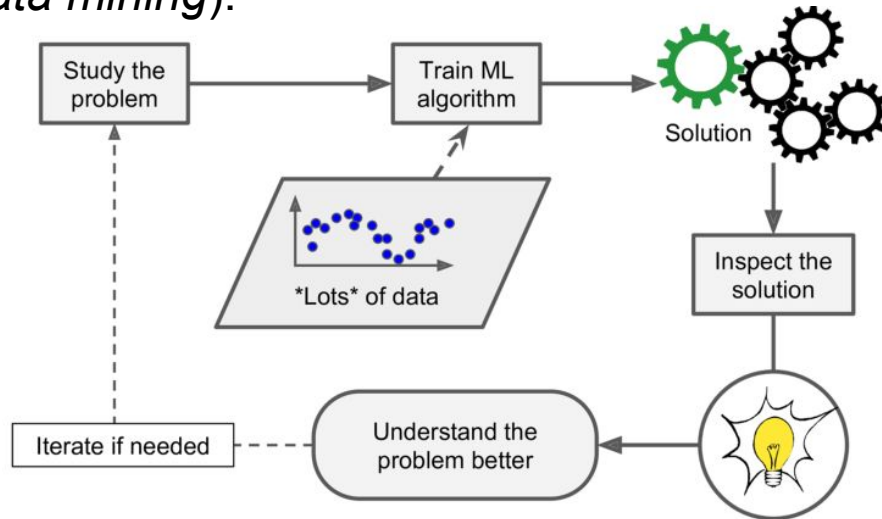
# Ejemplo: Machine Learning automatizado

- Es más fácil de mantener (y más preciso).
- Entender qué un indicador de spam en el texto cambio de “4U” a “U” es más rápido con este enfoque.

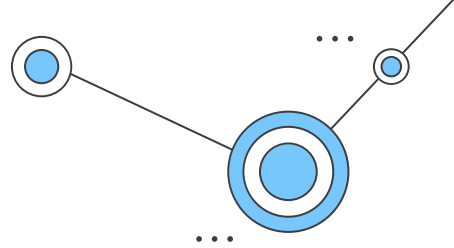


# Machine Learning en problemas complejos

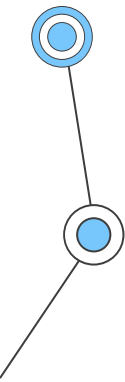
- Ej: Reconocimiento del habla.
- A partir de la inspección de los modelos de Machine Learning se podría entender qué patrones está reconociendo en los datos y extraer conocimiento (*data mining*).



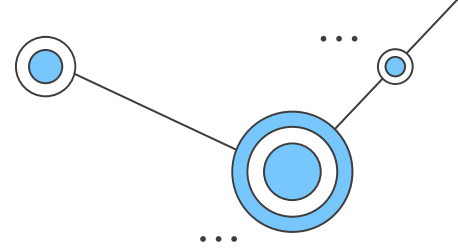
# Ejemplos de aplicaciones del Machine Learning



- Analizar imágenes de productos en la línea de producción para clasificarlos (CNN).
- Detectar tumores en escaneos cerebrales. (CNN)
- Clasificación automática de artículos. (NLP: RNN or Transformers)
- Etiquetado de comentarios ofensivos (NLP).
- Pronóstico de métricas de una compañía. (Linear or Polynomial regression, Random Forest, SVM or ANN)
- Detecting credit card fraud (detección de anomalía)
- Segmentar clientes en base a su compras para utilizar distintas campañas de marketing en ellos (clustering)
- Representar datos de alta dimensionalidad en un diagrama (reducción de dimensionalidad)
- Construir un bot para jugar un juego (reinforcement learning)

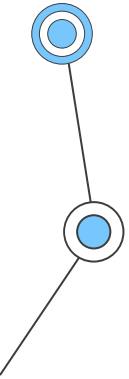


# Tipos de sistemas de Machine Learning



Se pueden categorizar a los sistemas de Machine Learning por los siguientes criterios:

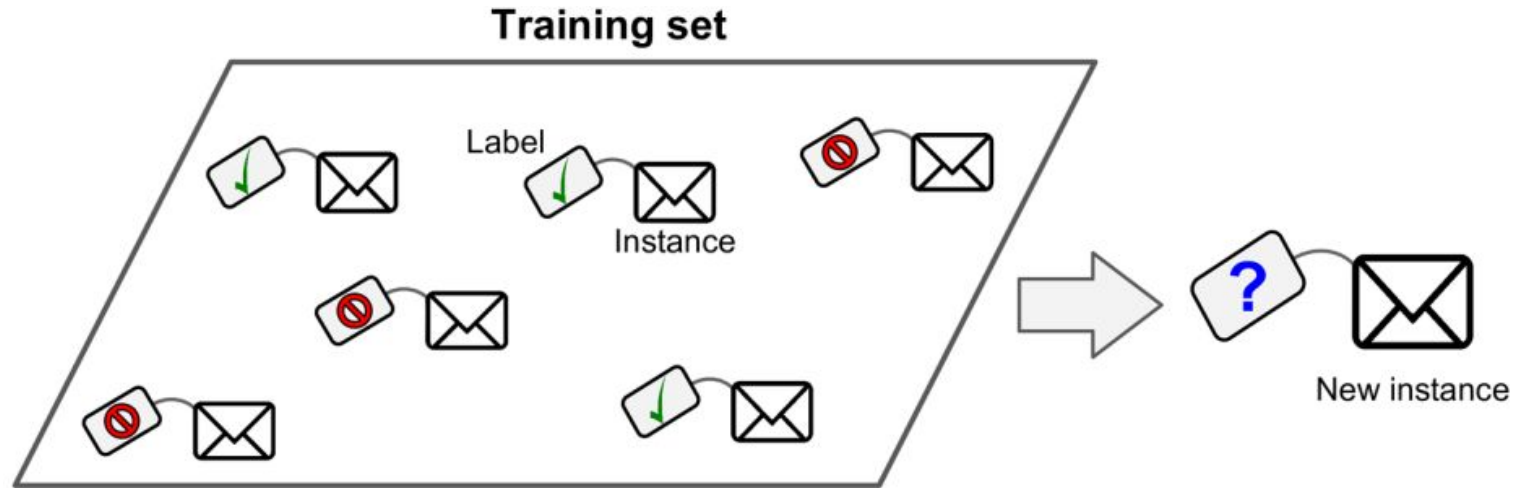
- Si necesitan supervisión humana (supervisados, no supervisados, semi supervisados y reinforcement learning).
- Si pueden o no aprender incrementalmente (online y batch learning).
- Si se construyen de comparar instancias previas con instancias nuevas o tratan de generalizar un comportamiento modelando. (basado en instancia o basado en modelos)



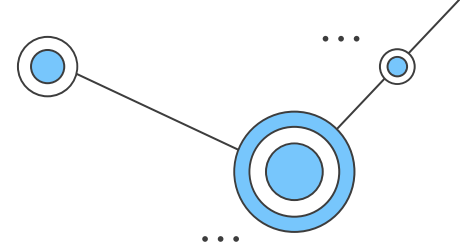
# Supervisado /no supervisado

# Supervisado

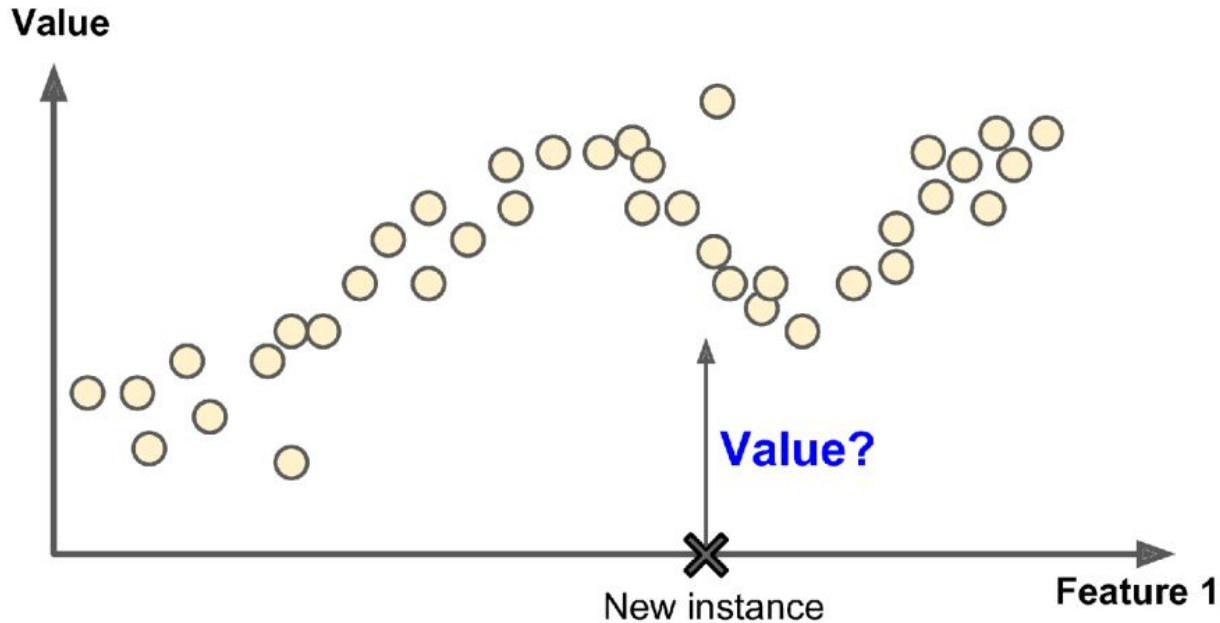
El conjunto de datos con los que se entrenan (training set) estos algoritmos necesitan estar etiquetados.



# Supervisado

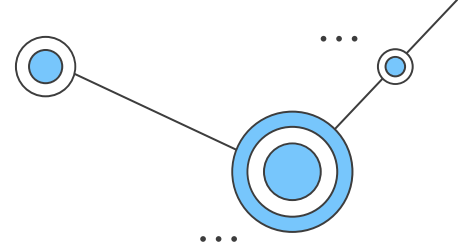


También puede referirse a valores numéricos



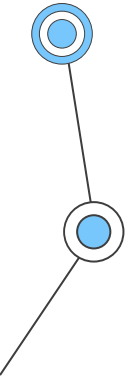


# Supervisado

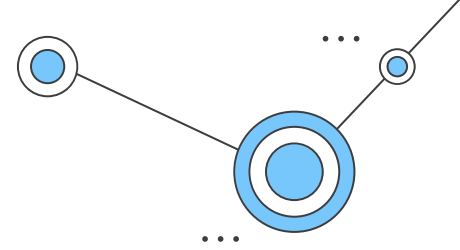


Algoritmos supervisados:

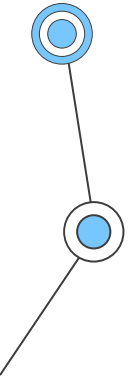
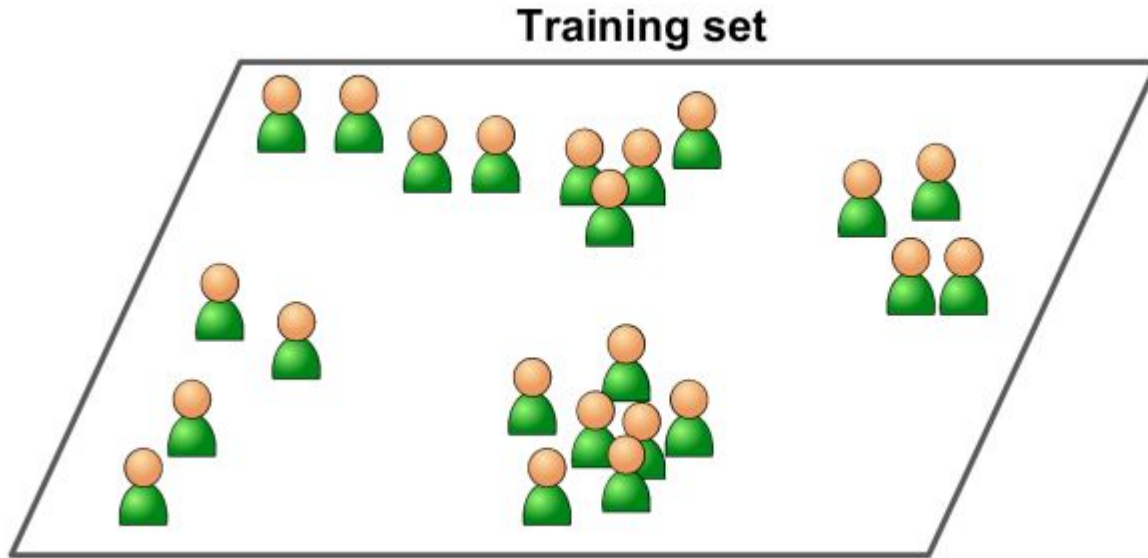
- k-Nearest Neighbors (kNN)
- Regresión Lineal.
- Regresión Logística.
- Support Vector Machines (SVM)
- Decision Trees y Random Forest.
- Redes Neuronales.



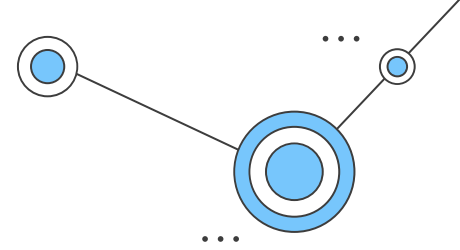
# No supervisado



En este caso el conjunto de entrenamiento no está etiquetado.

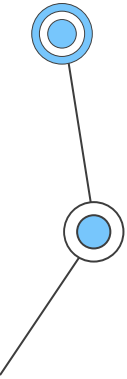


# No supervisado

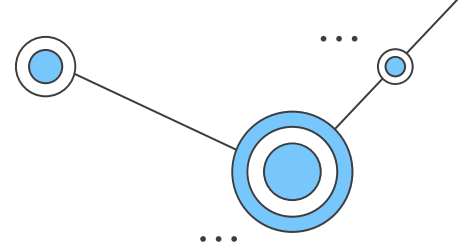


Algunos de los algoritmos no supervisados.

- Agrupamiento:
  - K-Means
  - DBSCAN
  - Agrupamiento jerárquico (HCA)
- Detección de anomalía
  - One-class SVM
  - Isolation Forest

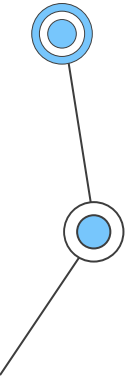


# No supervisado

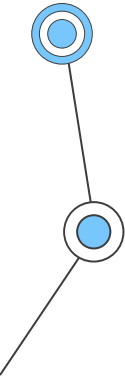
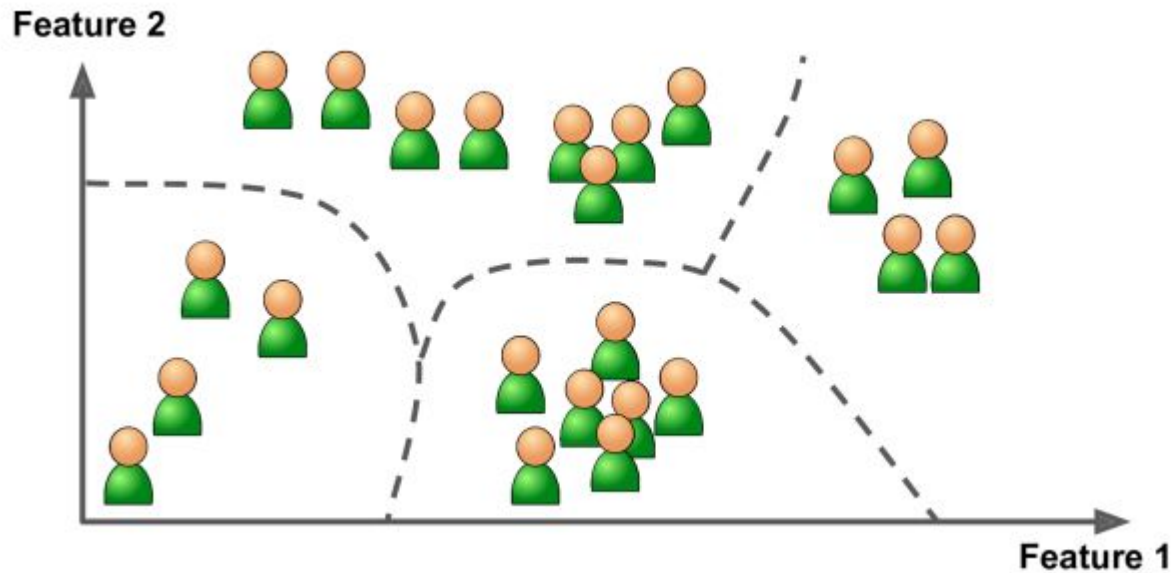
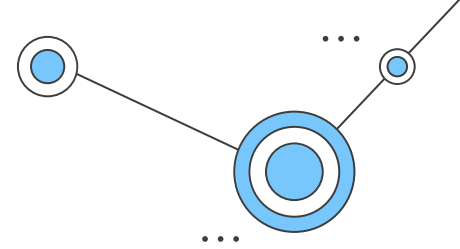


Algunos de los algoritmos no supervisados.

- Visualización y reducción de dimensionalidad:
  - Análisis de componentes principales (PCA)
  - Kernel PCA.
  - Locally Linear Embedding (LLE)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE)
  - Self-organizing map (SOM)
- Aprendizaje reglas de asociación
  - Apriori
  - Eclat



# No supervisado: Agrupamiento



# No supervisado: algoritmos de visualización

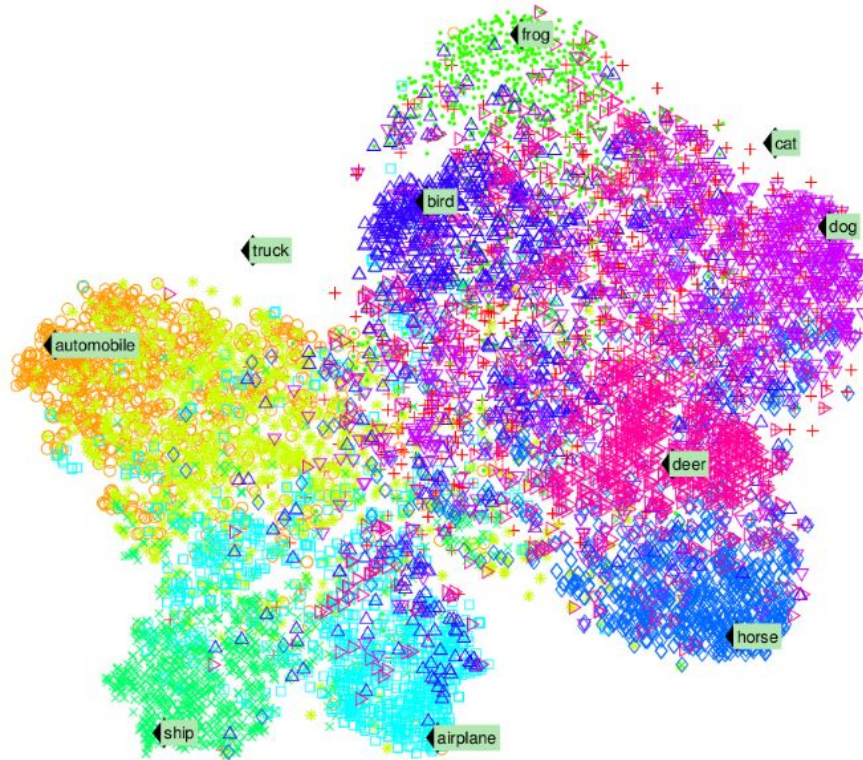
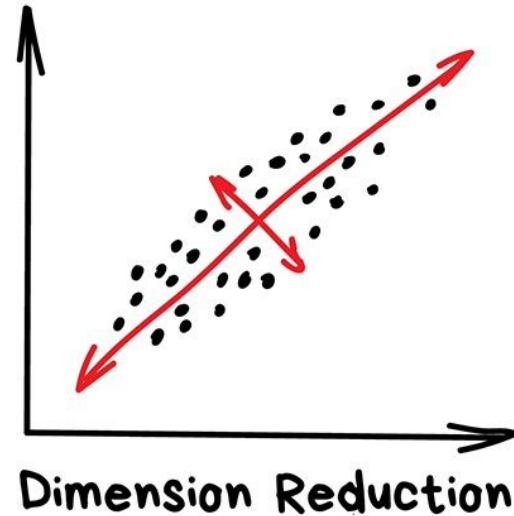


Figura de Richard Socher et al., "Zero-Shot Learning Through Cross-Modal Transfer," Proceedings of the 26th International Conference on Neural Information Processing Systems 1 (2013): 935–943.

# No supervisado: Reducción de dimensionalidad

- Reducir la cantidad de información sin perder mucha información relevante.
- El algoritmo de reducción de dimensionalidad va a tratar de combinar varias características (feature) en una. Esto se denomina extracción de características (feature extracción)



# No supervisado: Detección de anomalía o de novedad

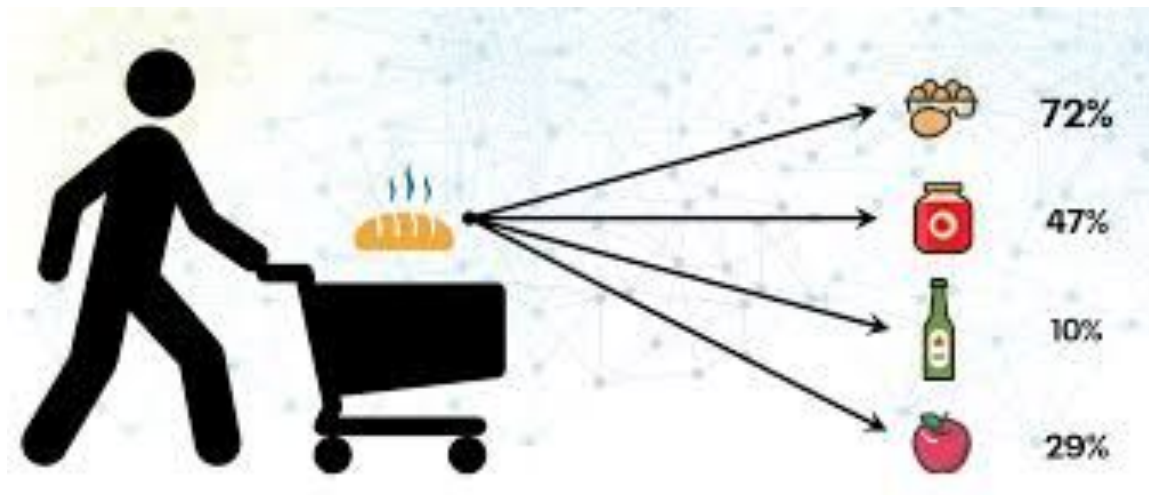
- Son algoritmos que se entrenan con instancias normales y detectan si existen anomalías en las nuevas instancias.
- Ej: fraude en transacciones de tarjeta de crédito, defectos en manufacturación o remover anomalías en un dataset antes de utilizarlo con otro algoritmo.





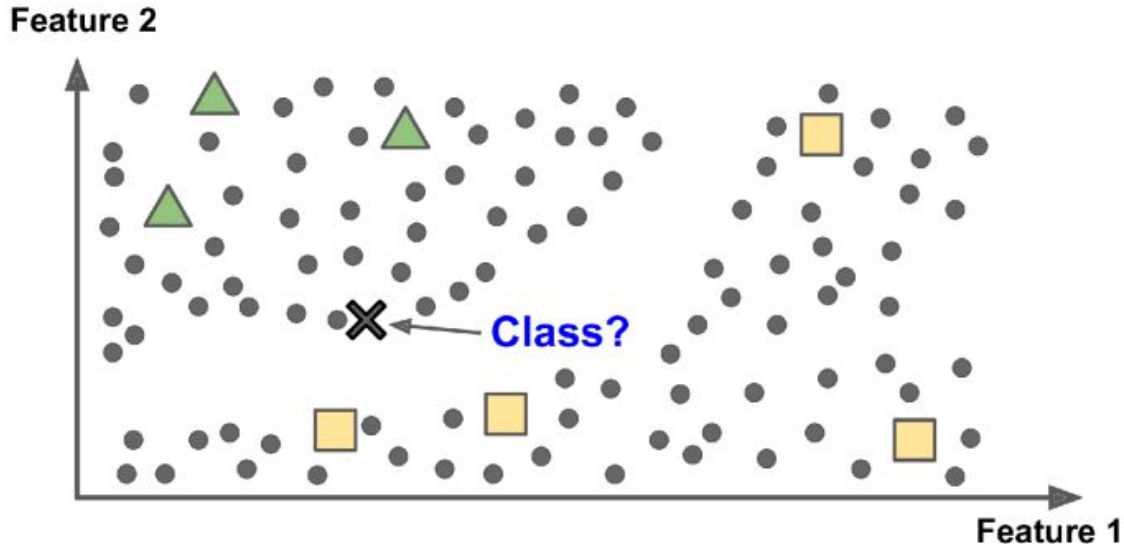
# No supervisado: Reglas de asociación

Como su nombre indica, su objetivo es entrar en grandes cantidades de datos relacionales entre los atributos.



# Semi Supervisado

Algunos algoritmos pueden funcionar con datos parcialmente etiquetados.  
Ej: deep belief networks (DBNs) basadas en restricted Boltzmann machines (RBMs). Se entrenan de una manera no supervisada y después sobre todo el sistema se realiza un fine-tuning.

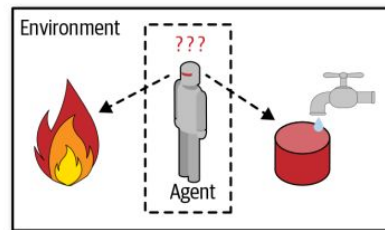


# Reinforcement Learning

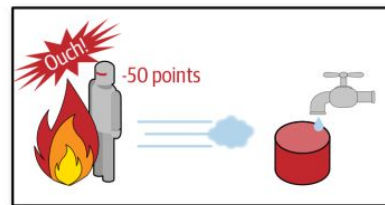
Sistema de aprendizaje (agente) en un contexto (ambiente) puede desarrollar acciones y recibir recompensa como retornos.

Aprende por sí mismo cuál es la mejor estrategia a seguir (política) para ganar más recompensa en el tiempo.

Ej: [DeepMind's Alpha Go](#), [OpenAI Five](#)



- 1 Observe
- 2 Select action using policy



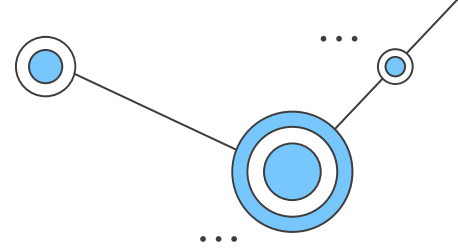
- 3 Action!
- 4 Get reward or penalty



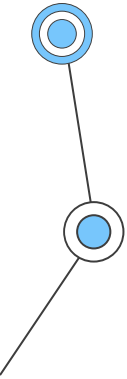
- 5 Update policy (learning step)
- 6 Iterate until an optimal policy is found

# Batch o online learning

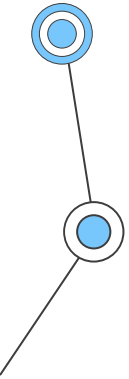
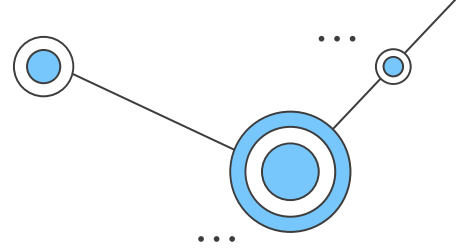
# Batch learning



- No aprende de forma incremental: debe ser entrenado con todos los datos disponibles. También se denomina offline.
- Con nuevos datos, todo el sistema debe entrenarse desde cero.
- Los procesos de entrenamiento, evaluación y desarrollo pueden automatizarse.
- No sirve para sistemas que necesitan adaptarse rápido (algunos sistemas pueden tomar más de un día para entrenarse).
- Pueden ser muy costosos computacionalmente.

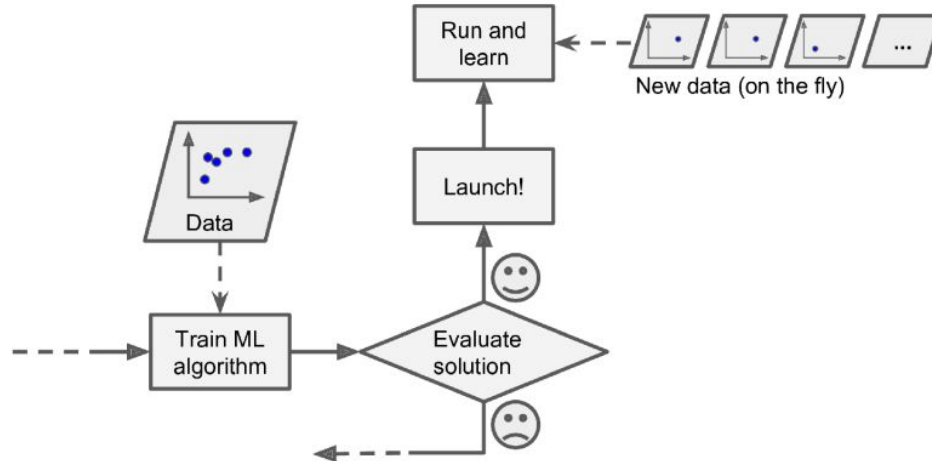


# Online learning



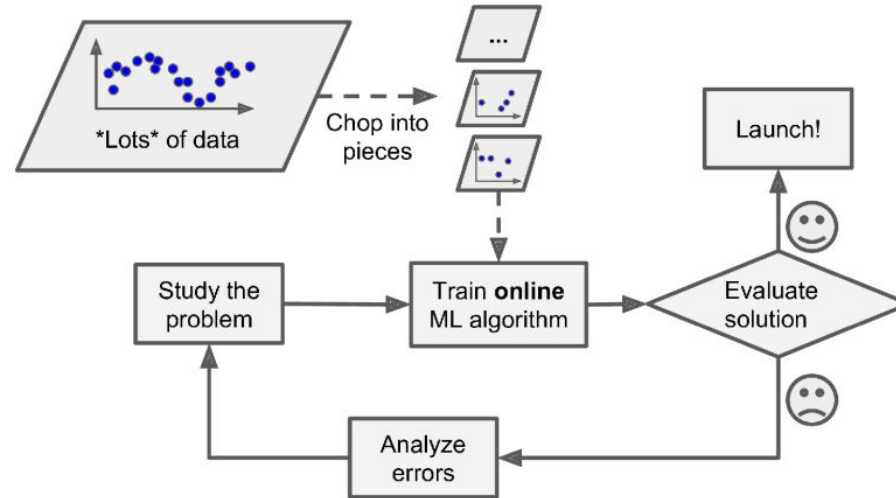
# Online learning

- Aprende de forma incremental: Se alimenta de datos secuencialmente, puede ser individuales o en pequeños (mini-batches).
- Cada entrenamiento del algoritmo es “barato”.
- Se utilizan para sistemas que reciben información permanentemente y necesitan adaptarse a los cambios.



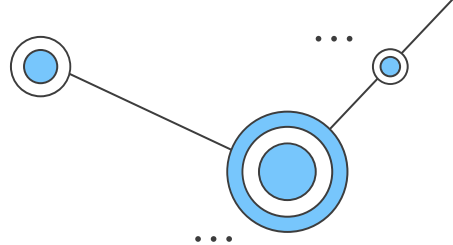
# Online learning

- Se utiliza también cuando la cantidad de datos es tan grande que no cabe en un servidor (out-of-core learning).

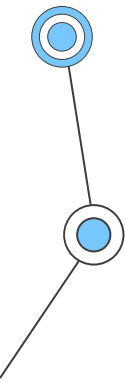






# Online learning

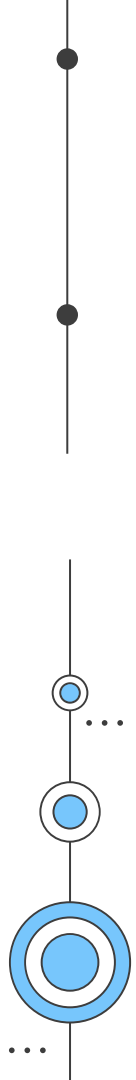


- Learning rate: es qué tan rápido mi algoritmo se adapta a los cambios.
- Learning rate alto trata de adaptarse más rápidamente a los nuevos datos. Learning rate bajo se adapta más despacio.
- **Nota: Si los nuevos datos son malos tu sistema podría empezar a bajar su performance!**
- Es necesario monitorear el modelo en caso de que pase lo anterior para poder apagarlo y volver a un modelo anterior.



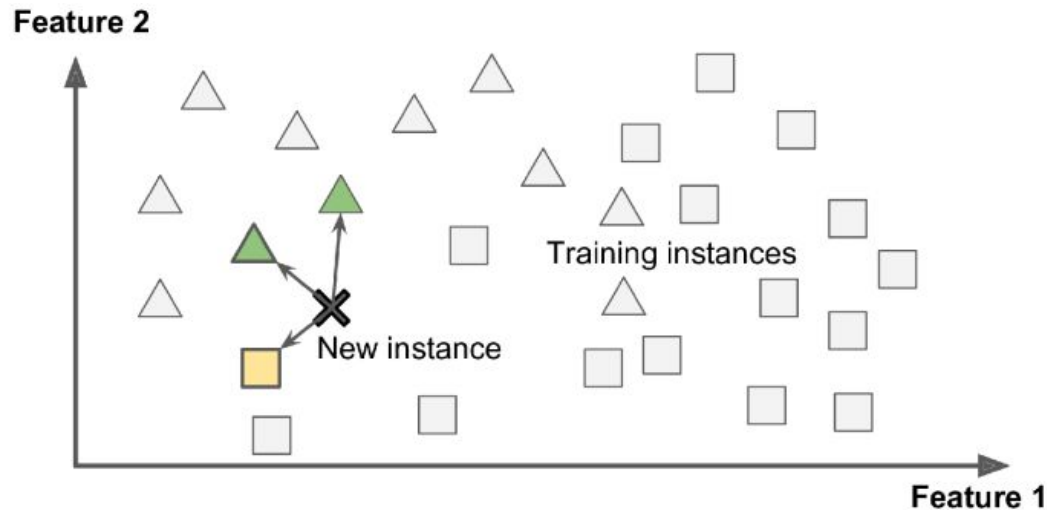


**Basado en  
instancia o Basado  
en modelos**



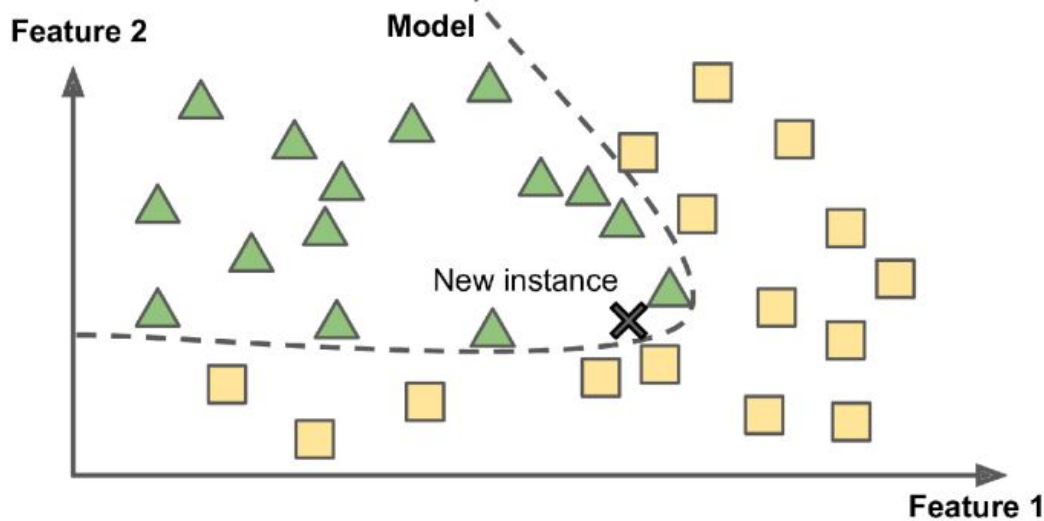
# Aprendizaje basado en instancia

- Aprende “de memoria”
- Utiliza una medida de similitud para comparar nuevos casos con los casos aprendidos con anterioridad.

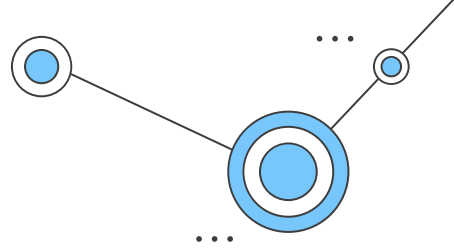


# Aprendizaje basado en modelo

- Construir un modelo para realizar predicciones.



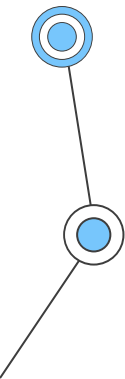
# Aprendizaje basado en modelo: Ejemplo



Queremos saber si un mayor PIB hace más felices a las personas

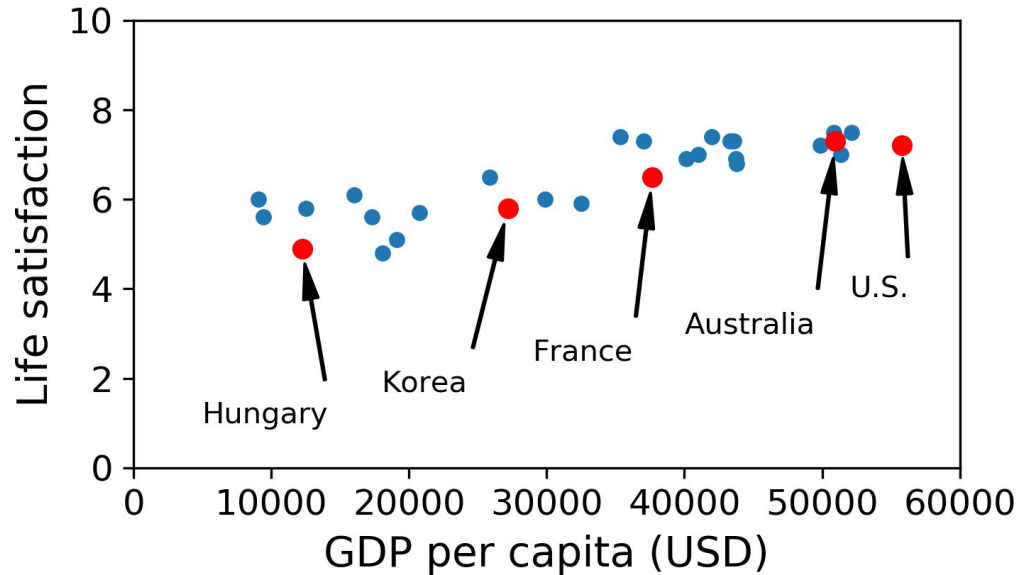
*Table 1-1. Does money make people happier?*

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2



# Aprendizaje basado en modelo: Ejemplo

Graficamos:

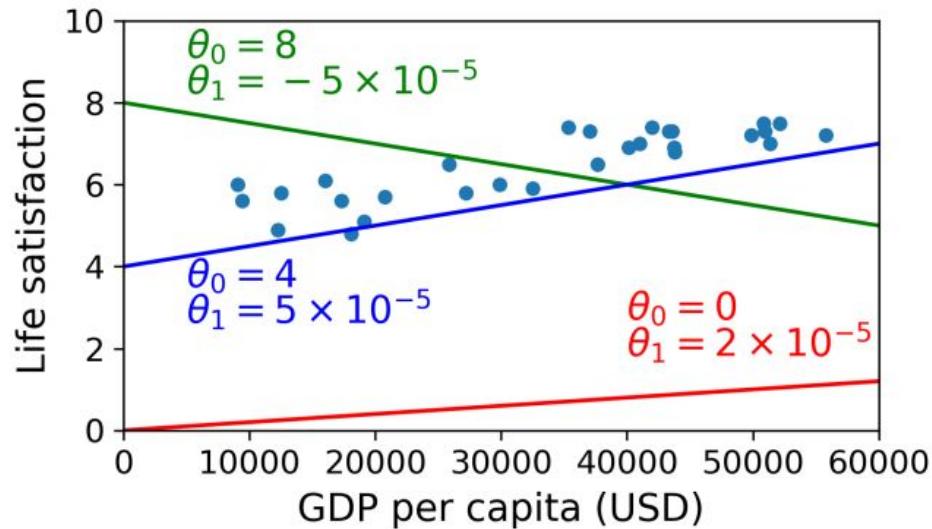


Construimos un modelo:

$$\text{life\_satisfaction} = \theta_0 + \theta_1 \times \text{GDP\_per\_capital}$$

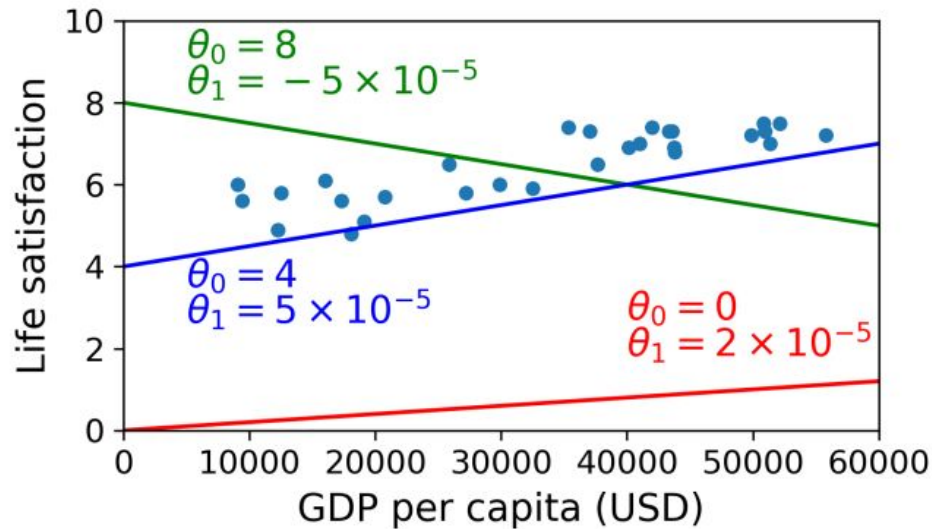
# Aprendizaje basado en modelo: Ejemplo

¿Cómo podemos saber qué modelo es mejor?



# Aprendizaje basado en modelo: Ejemplo

¿Cómo podemos saber qué modelo es mejor?

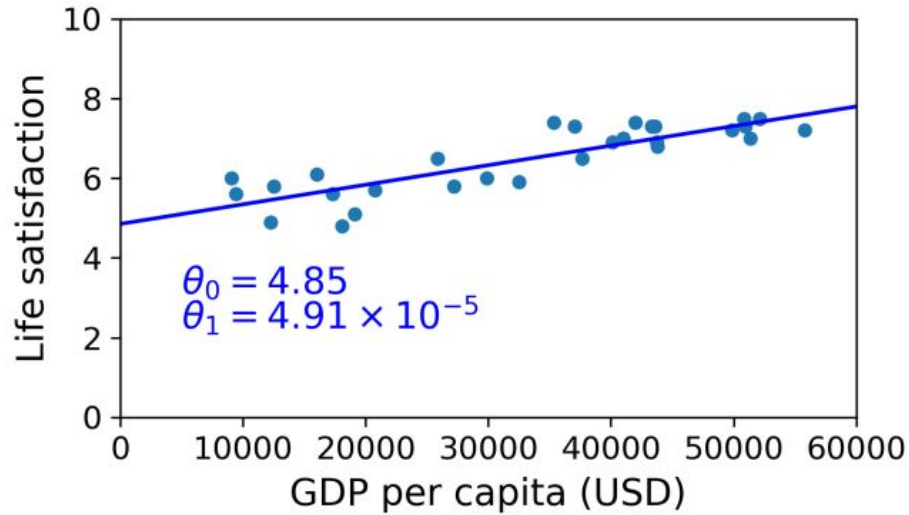


En regresión lineal se utiliza una medida de distancia entre la línea del modelo y los datos.



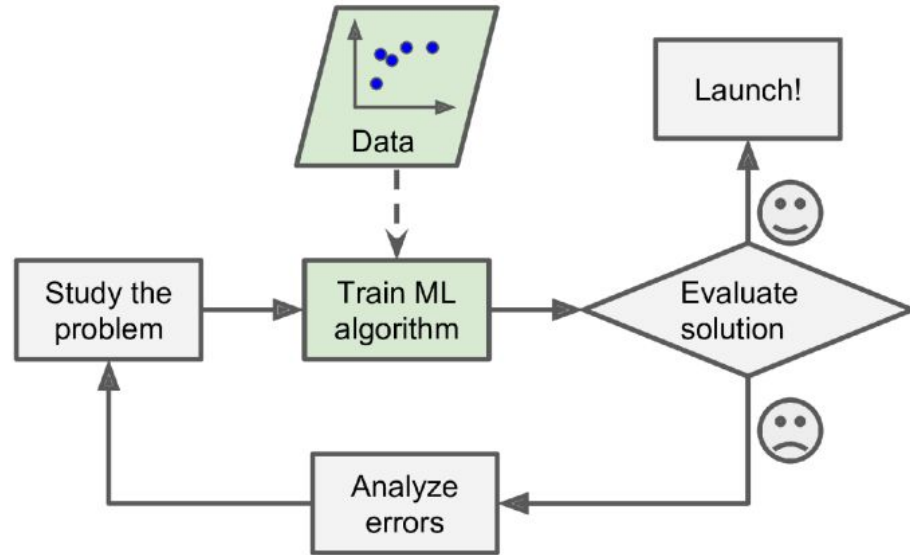
# Aprendizaje basado en modelo: Ejemplo

Se ajusta el modelo a los datos

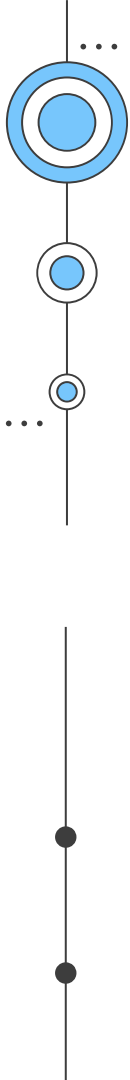


# Ejemplo: Machine Learning

También pueden ser fácilmente automatizados.

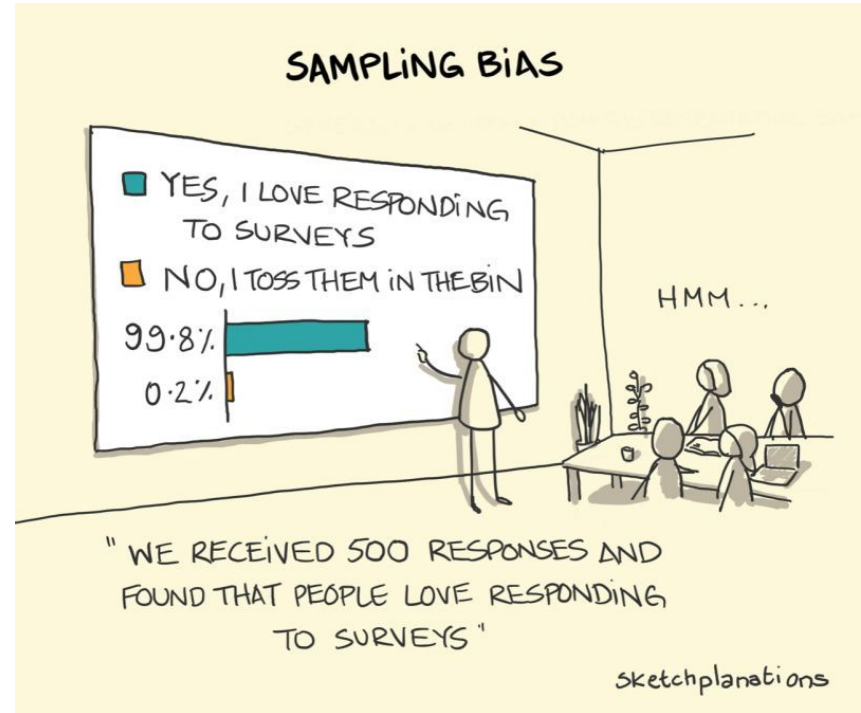


# Principales retos del Machine Learning



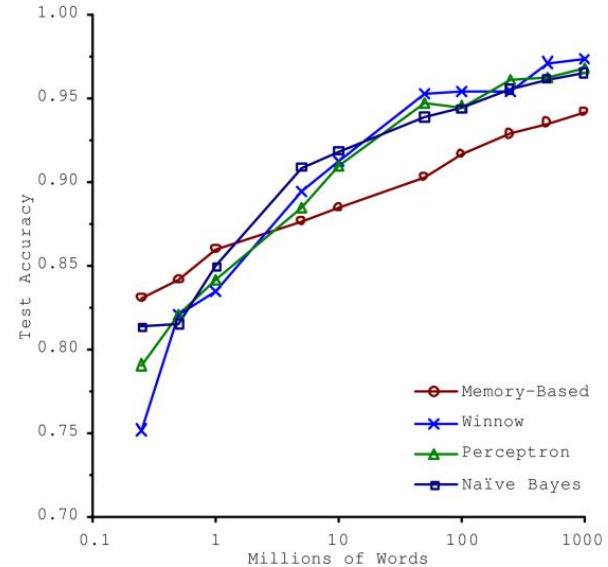
# La importancia buenos datos

si los datos son malos, incluso el mejor algoritmo no te ayudará.  
“basura entra, basura sale”  
(GIGO).



# Cantidad insuficientes de datos

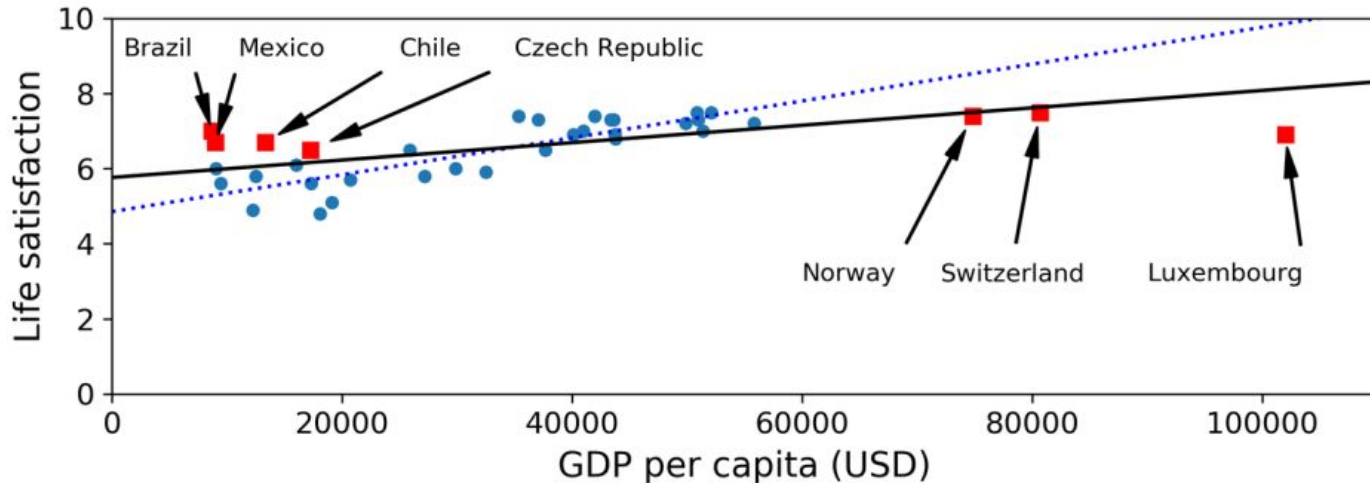
- Algunos algoritmos del ML necesitan mucha cantidad de datos.
- Trade-off entre tiempo y dinero gasta en algoritmos y recolección de datos.



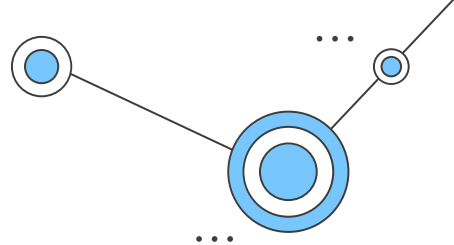
“Scaling to Very Very Large Corpora for Natural Language Disambiguation” Michele Banko and Eric Brill, 2001

# Datos no representativos

- Si la muestra no es representativa el modelo no va a generalizar bien.
- La muestra puede ser no representativa porque es muy pequeña y estaríamos modelando ruido.
- Incluso en muestras muy grandes puede haber un sesgo de muestreo.



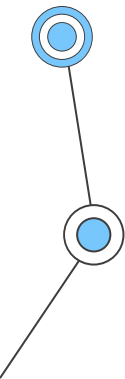
# Baja calidad de datos



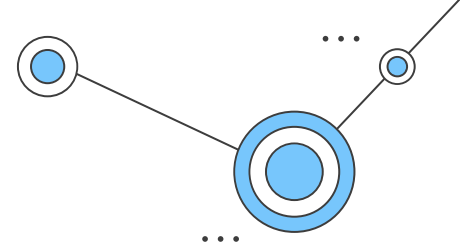
Si nuestro training set contiene errores, valores atípicos o ruido; es muy difícil que el modelo tenga un buen desempeño.

Algunas acciones a realizar para limpiar los datos pueden ser:

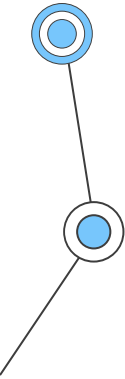
- Si existen valores atípicos, se puede simplemente descartar estos datos (edad de clientes mayor a 120 ).
- Si algunas características de la muestra no está cargada (el 5% de los clientes no especificaron su edad), se debe decidir si ignorar esas instancias, llenar los valores faltantes o directamente no utilizar esta característica.



# Características irrelevantes



- Los sistemas van a ser capaces de entrenarse y desempeñar bien su tarea sólo si se contiene suficientes features relevantes.
- Una parte importante del entrenamiento es entrenar el sistema con una buen conjunto de características, a esto se lo denomina feature engineering. Pasos:
  - Selección de características
  - Extracción de características
  - Creación de nuevas características almacenando más datos.



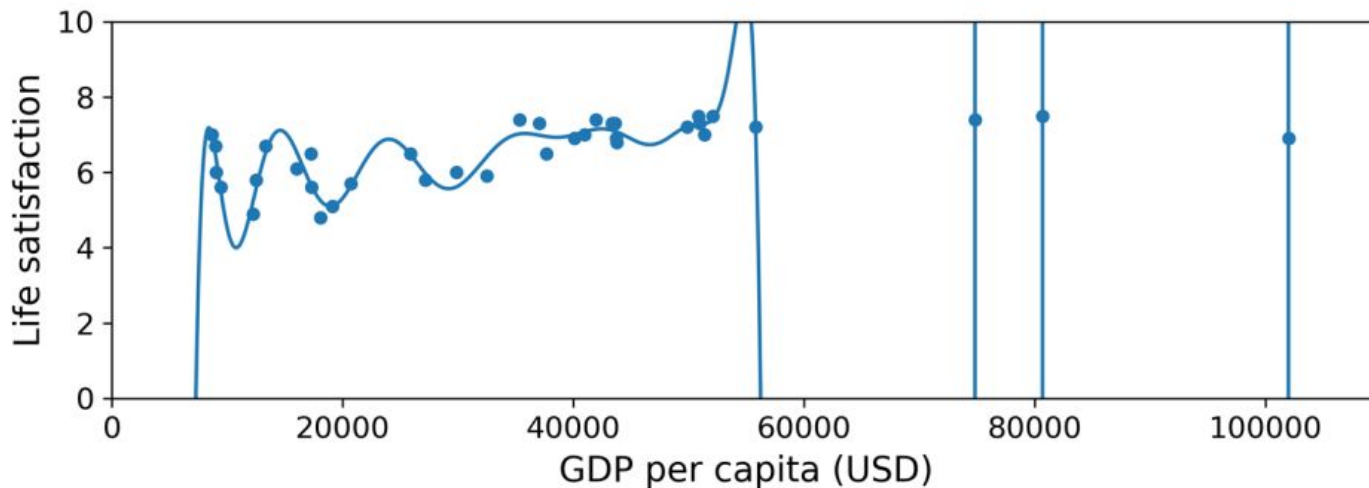


# Sobreajuste (overfitting) de los datos de entrenamiento

- Modelo muy complejo para la cantidad de datos y ruido.

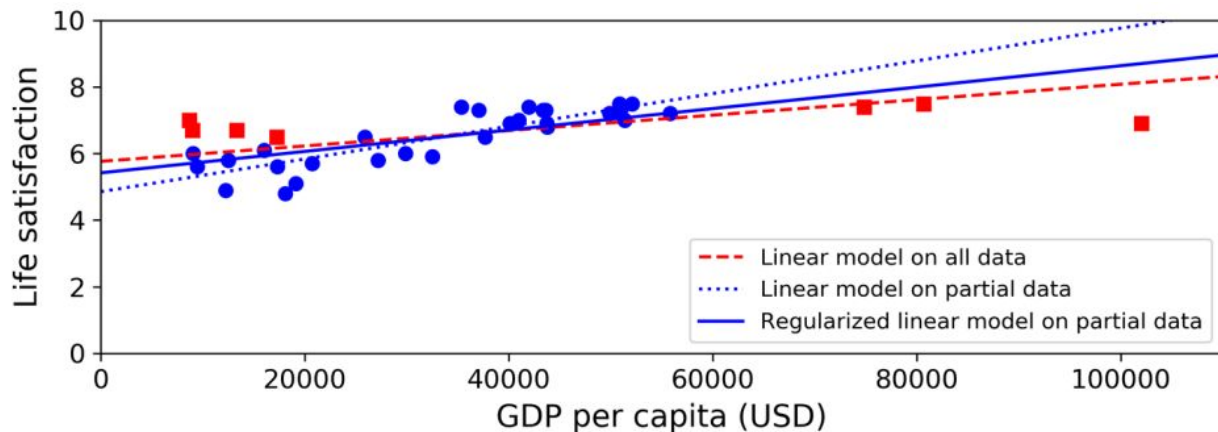
Posibles soluciones:

- Simplificar el modelo.
- Reunir más datos.
- Reducir el ruido.

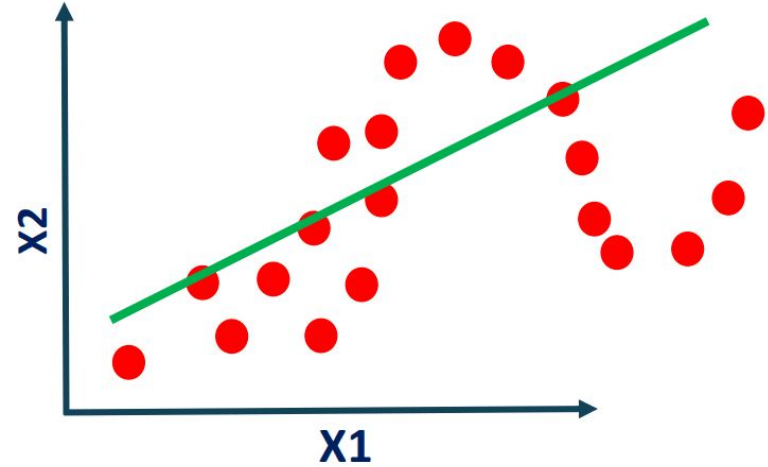


# Sobreajuste (overfitting) de los datos de entrenamiento

- Regularización: reducir los grados de libertad del modelo.
- La cantidad de regularización de un modelo se controla con un hiperparámetro. Más grande su valor el modelo empieza a capturar cada vez menos del comportamiento de los datos.

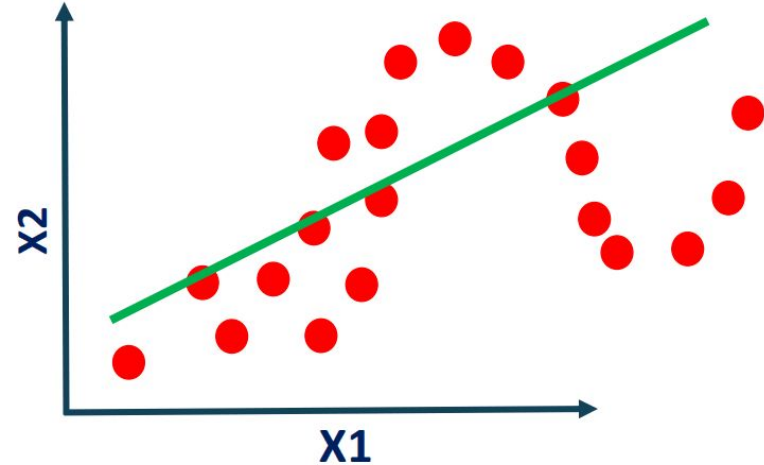


# Subajuste (underfitting) de los datos de entrenamiento



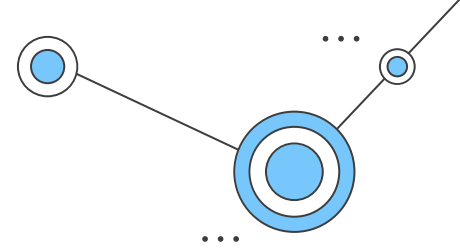
# Subajuste (underfitting) de los datos de entrenamiento

- Modelo muy simple para modelar los datos.
- Posibles soluciones:
  - Seleccionar modelos más complejos.
  - Realizar feature engineer.
  - Reducir las restricciones del modelo (reducir la regularización de hiperparámetros)

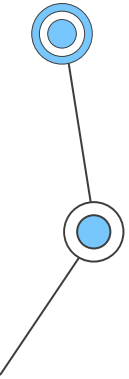


# Evaluación y Validación

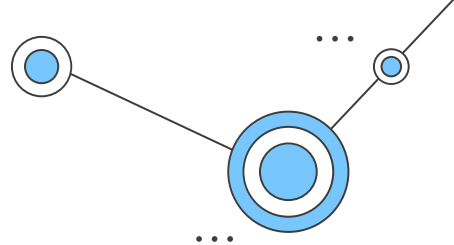
# Validación y evaluación



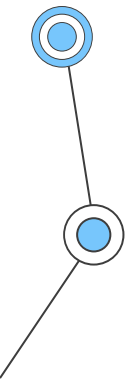
- ¿Es buena idea poner a funcionar un modelo en producción y monitorearlo?



# Validación y evaluación

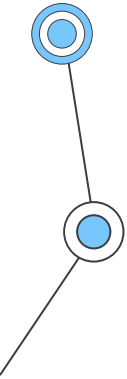
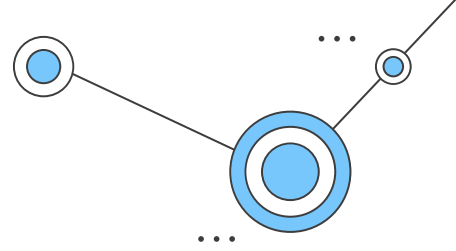


- ¿Es buena idea poner a funcionar un modelo en producción y monitorearlo? NO!
- Utilizar training set y test set (generalmente la proporción que se utiliza es el alrededor de 80% - 20%)
- El error en el test set se denomina error out-of-sample o error de generalización.



# Ajuste de hiperparámetros y selección de modelo

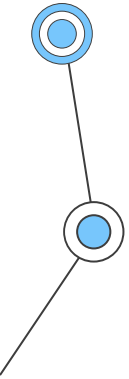
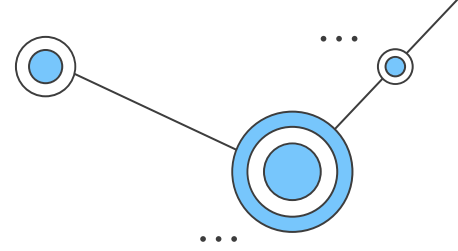
- ¿Cómo puedo comparar entre dos modelos?





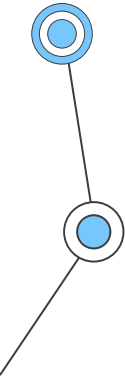
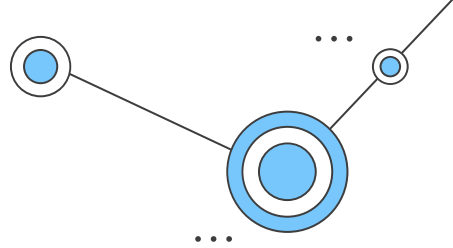
# Ajuste de hiperparámetros y selección de modelo

- ¿Cómo puedo comparar entre dos modelos?
- Utilizar el error de generalización en el test set, pero...



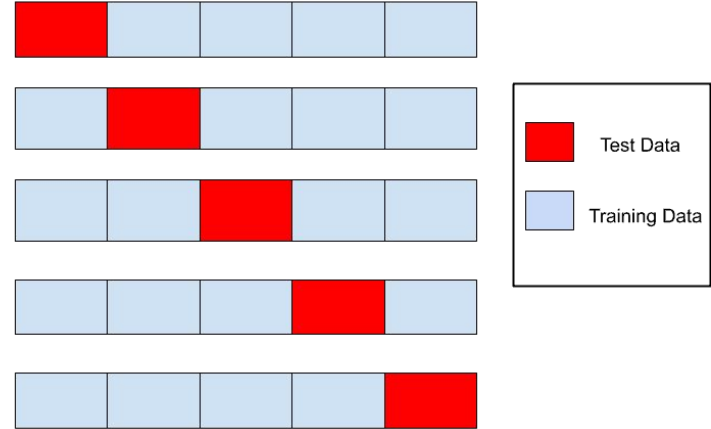
# Ajuste de hiperparámetros y selección de modelo

- ¿Cómo puedo comparar entre dos modelos?
- Utilizar el error de generalización en el test set, pero podría obtener un error mayor una vez que lleve el modelo a producción.

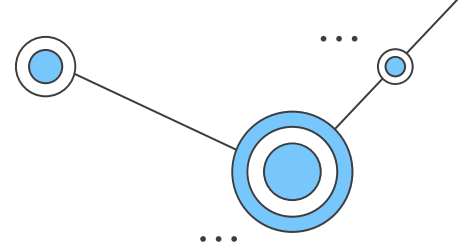


# Ajuste de hiperparámetros y selección de modelo

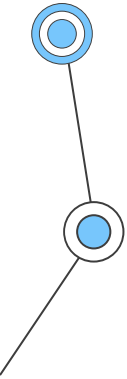
- Para solucionar lo anterior se usa validación cruzada. Se vuelve a dividir el training set en training set y validation set.
- El punto anterior se puede hacer muchas veces para obtener valores distintos. Esto se lo denomina validación cruzada.



# No free lunch theorem



- Publicado en 1996 por David Wolpert.
- Si no se sabe absolutamente nada de los datos no puedes elegir un modelo sobre otro a priori.
- En la práctica vamos a tener que realizar suposiciones sobre los datos y evaluar unos pocos modelos.



# Repaso

# Algunas preguntas de repaso

- Que es un etiquetado en el conjunto de entrenamiento.
- ¿En qué categoría entraría algoritmo sería la detección el algoritmo de detección de spam? supervisado o no supervisado.
- ¿Cual es la diferencia entre un parámetro del modelo y un hiperparámetro?
- Si el modelo funciona muy bien en el conjunto de entrenamiento, pero mal en el evaluación ¿qué está pasando?
- ¿Para qué utilizo el conjunto de validación?
- ¿Se pueden ajustar los hiperparámetros en el test set?