


2023/


# ITBA

# Maestría en Management & Analytics


## Clase 3 - Introducción a Machine Learning



En esta clase vamos a realizar una introducción a **Machine Learning** repasando su definición y planteando algunos de los conceptos principales que nos servirán para el curso.



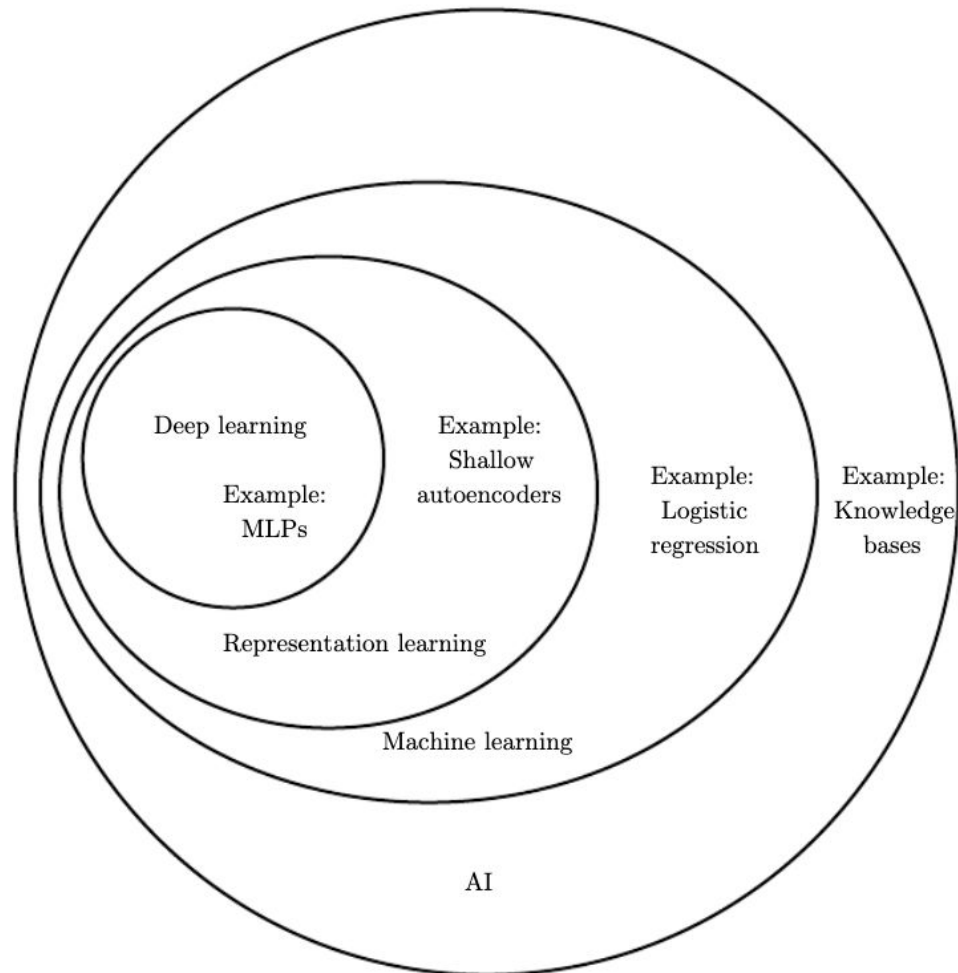
Además presentaremos la librería **Scikit-learn** y realizaremos una primera demo sobre cómo implementarla en un problema de regresión.



Definición de Arthur Samuel, 1959:

"[El aprendizaje automático es el] campo de estudio que da a los ordenadores la capacidad de aprender sin ser programados explícitamente."

# Definición de Machine Learning



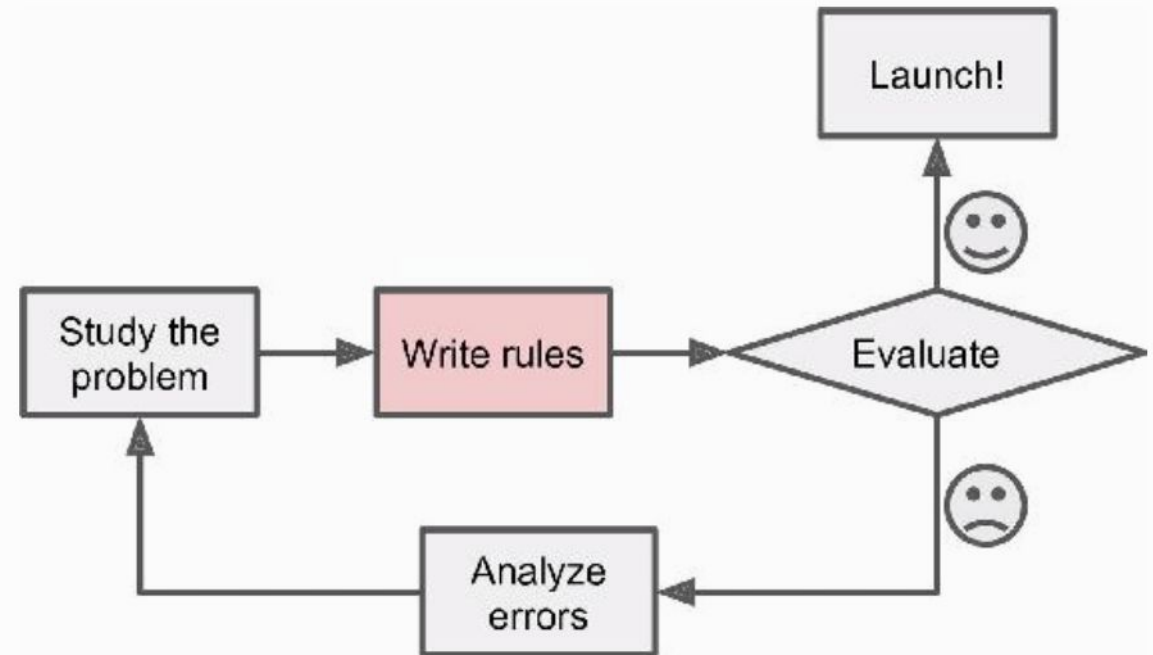
**Machine Learning** es un subcampo de la IA, en el cual los **sistemas son entrenados** a partir de datos, en lugar de ser programados explícitamente.

# Sistemas basados en reglas

Posible aplicación de un **sistema basado en reglas** → diagnósticos médicos

1. Consultar expertos y libros de texto.
2. Representar explícitamente el conocimiento de los expertos como una serie de instrucciones.
3. Escribir un algoritmo que implemente esas instrucciones.

Enfoque muy popular en los años '80



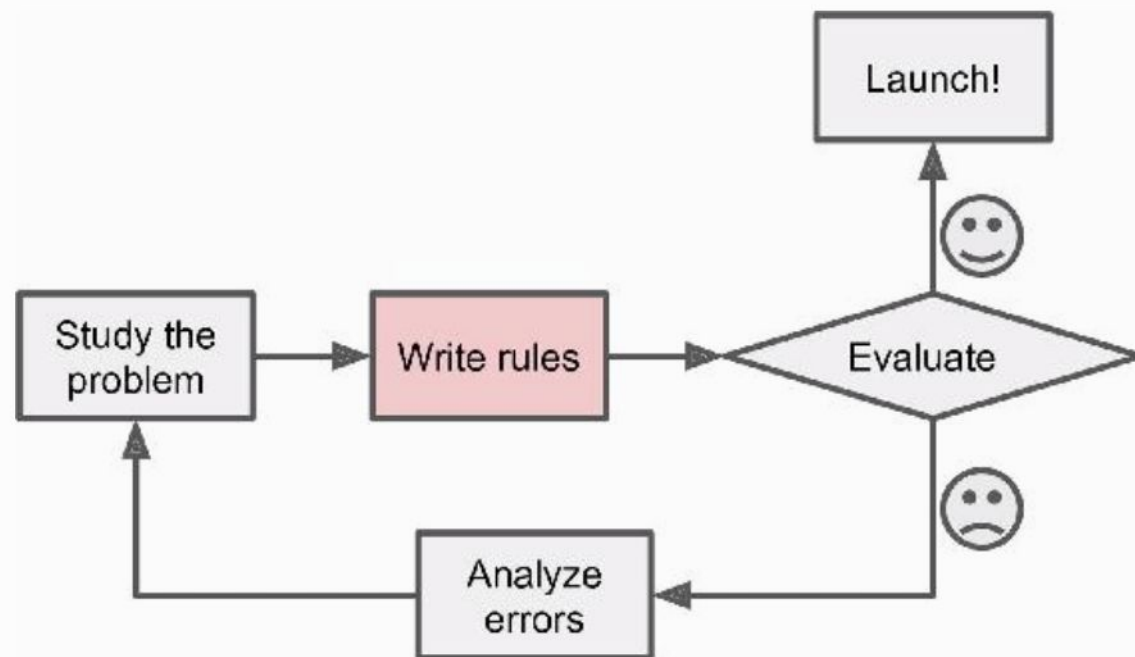
Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. - Aurélien Géron (2017)

# Sistemas basados en reglas

## Problemas de los sistemas basados en reglas

1. Dificultad para escribir el set de instrucciones adecuado.
2. En particular, dificultad de representar problemas más complejos como clasificación de imágenes, traducciones de idiomas y reconocimiento del habla.
3. Dificultad de actualizar los sistemas

Las dificultades de este enfoque llevaron a un “invierno de IA” a inicios de los años ‘90.



Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. - Aurélien Géron (2017)

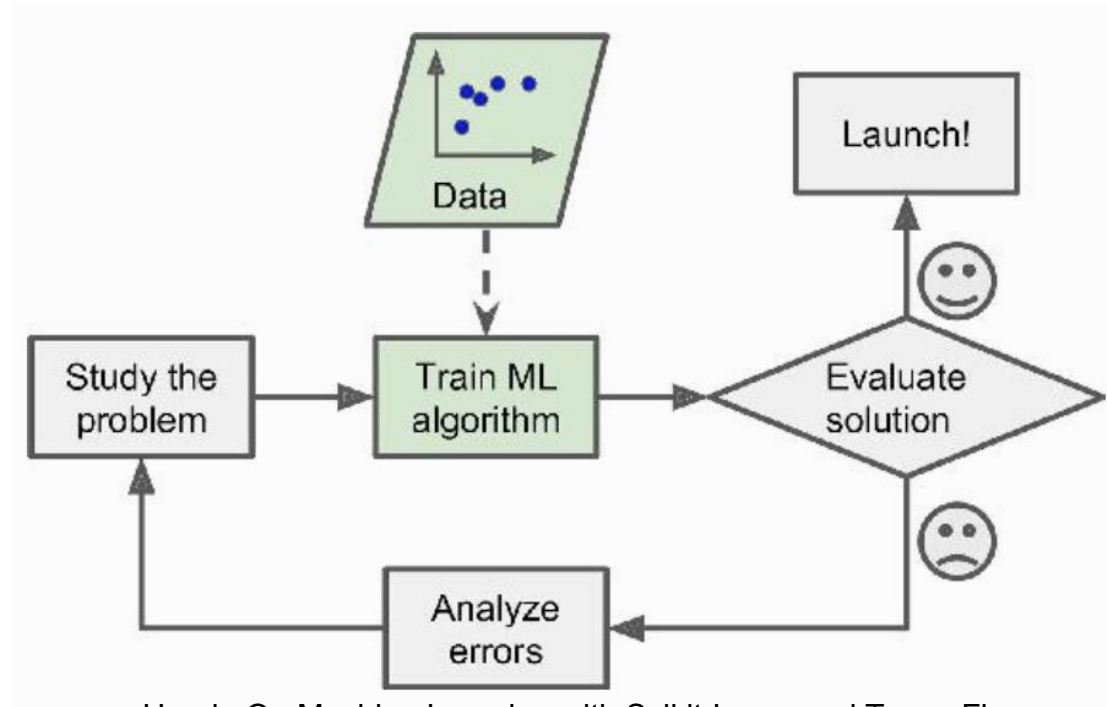
Definición de Tom Mitchell, 1997:

"Se dice que un programa de ordenador aprende de la experiencia  $E$  con respecto a alguna tarea  $T$  y alguna medida de rendimiento  $P$ , si su rendimiento en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ ."



Enfoque de **machine learning**:

1. En lugar de codificar el conocimiento de expertos, diseñar sistemas que puedan “aprender” de los datos.
2. Se proveen datos de entrenamiento, es decir muchos pares  $(x, y)$ , donde  $x$  es un vector de inputs e  $y$  es el output.
3. Este tipo de machine learning se llama **aprendizaje supervisado**.



Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. - Aurélien Géron (2017)

### Más formalmente:

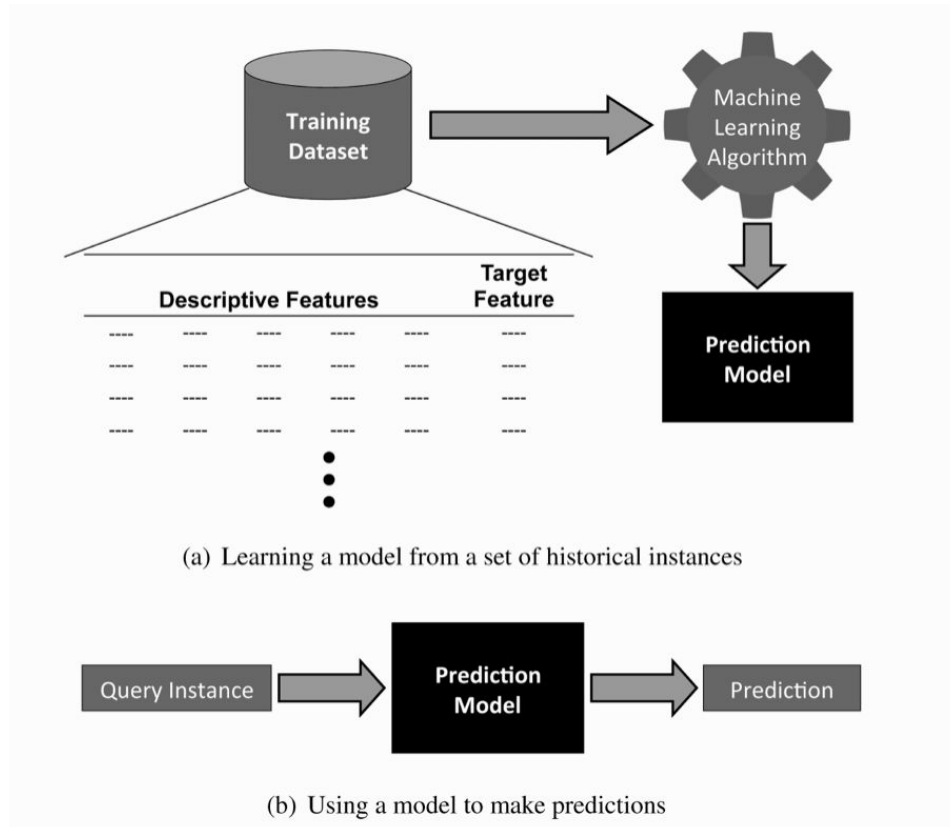
*Dados los datos de entrenamiento  $(\mathbf{x}, f(\mathbf{x}))$ ,  
donde  $\mathbf{x}$  es el input e  $y = f(\mathbf{x})$  es el output  
Encontrar una función  $h$  que aproxime a  $f$*

### Importante:

- Para que el modelo determinado por la función  $h$  sea útil, tiene que poder hacer predicciones para inputs que no están presentes en el set de entrenamiento.
- En ese caso, el modelo está capturando correctamente la relación subyacente entre el input y el output y por lo tanto se dice que el modelo **generaliza** bien.
- Para poder buscar a  $h$  se tiene que limitar el espacio en el que se buscan las hipótesis. Esto genera un **sesgo inductivo**.

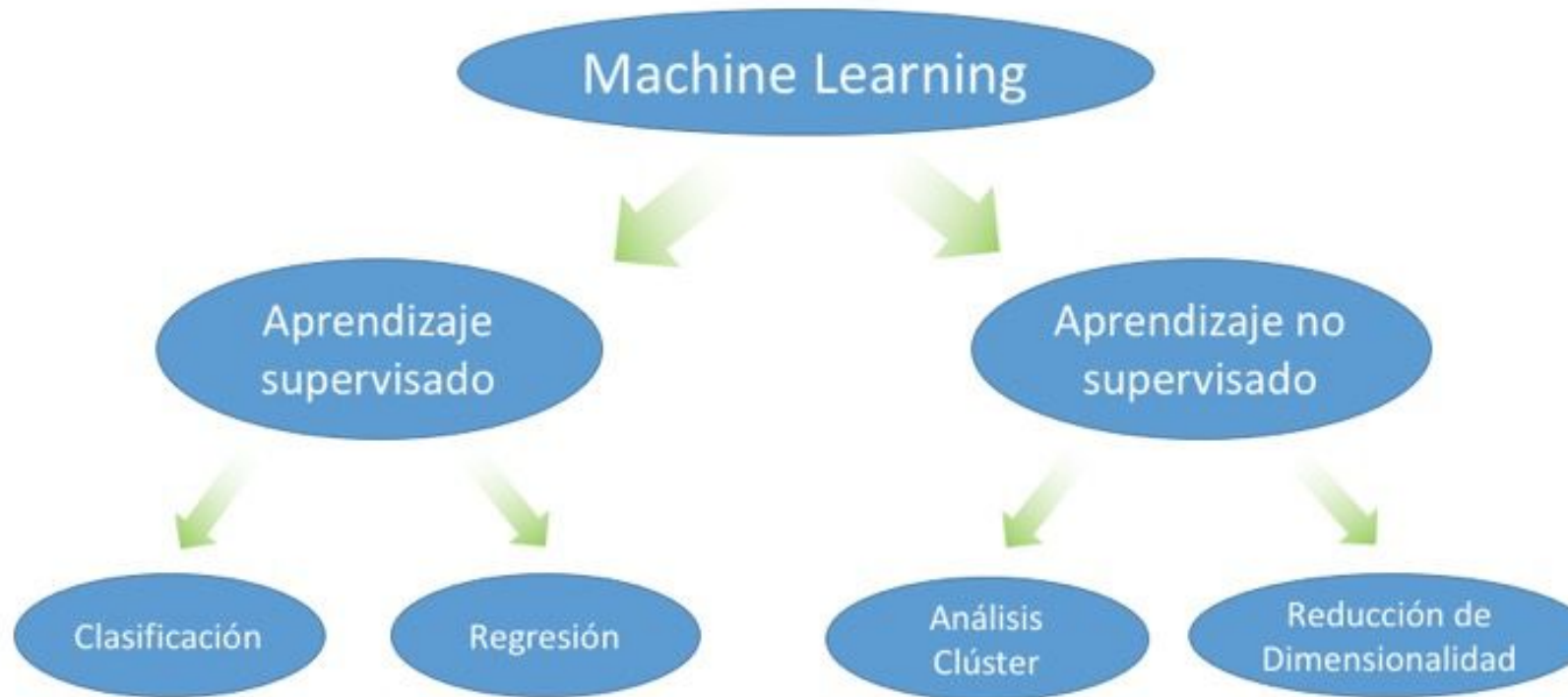
# Algoritmos de Machine Learning

En resumen, **machine learning** funciona buscando entre un conjunto de potenciales modelos (o **hipótesis**) para encontrar el modelo que mejor generaliza por fuera del set de entrenamiento. Los algoritmos de **machine learning** usan dos fuentes de información para guiar esta búsqueda: el **dataset de entrenamiento** y el **sesgo inductivo** asumido por el algoritmo.



“Fundamentals of Machine Learning for Predictive Data Analytics.” John D. Kelleher, Brian Mac Namee & Aoife D'Arcy(2020)

Clasificación de los algoritmos de Machine Learning:



Nota: el cuadro no contempla los algoritmos de aprendizaje semi-supervisado ni reinforcement learning

## Modelos paramétricos

- Aprenden de los datos mediante un conjunto fijo de parámetros.
- La cantidad de parámetros es independiente de la cantidad de datos.
- El algoritmo consiste en 2 pasos:
  - Definir la forma de la función
  - Aprender los coeficientes de la función a partir del set de entrenamiento

### Ventajas:

- Pueden ser rápidos de entrenar
- Pueden entrenarse con pocos datos (siempre y cuando se elijan funciones con pocos parámetros).

### Desventajas:

- Realizan fuertes supuestos sobre la naturaleza de los datos

## Modelos paramétricos: regresión lineal

Uno de los modelos paramétricos más utilizados para regresión es la regresión lineal. Este modelo asume que el output es aproximadamente una función lineal del input:

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \epsilon = \sum_{j=1}^D w_j x_j + \epsilon$$

Donde  $\mathbf{x}$  es el vector de inputs (o features) y  $\mathbf{w}$  es el vector coeficientes o parámetros.

Además, epsilon es el **error residual** entre nuestras predicciones lineales y el verdadero output. Asumimos que epsilon tiene una distribución:

$$\epsilon \sim \mathcal{N}(\mu, \sigma^2)$$

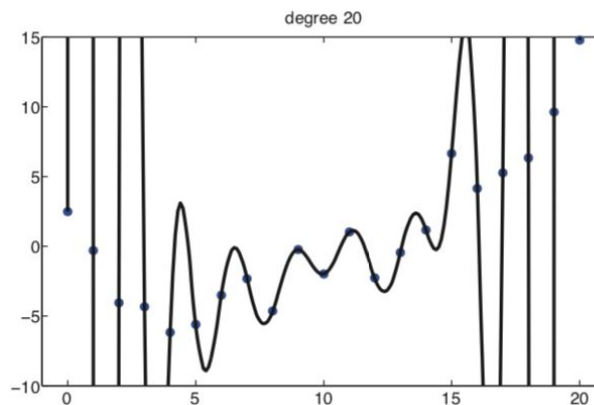
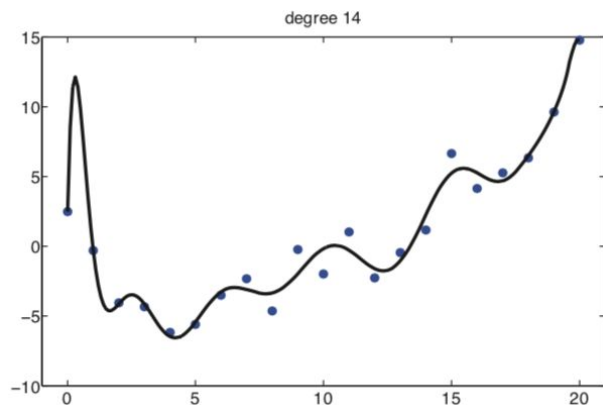
## Modelos paramétricos: regresión lineal

Los modelos de regresión lineal pueden capturar relaciones no lineales entre el input y el output, reemplazando al vector de inputs por una función no lineal de los inputs:

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \sigma^2)$$

Donde estamos haciendo explícita la relación entre la naturaleza aleatoria del output y la regresión lineal.

Por ejemplo tomando a  $\boldsymbol{\phi}(\mathbf{x}) = [1, x, x^2, \dots, x^d]$ , para  $d=14$  y  $d=20$ , obtenemos:



## Modelos no paramétricos

- No realizan supuestos fuertes sobre la forma de la función que mapea del input al output
- La cantidad de parámetros depende de las instancias (cantidad de datos) del dataset de entrenamiento.

### Ventajas:

- Son más flexibles
- Pueden tener mayor poder de ajuste

### Desventajas:

- Necesitan de gran cantidad de datos
- Más proclives a incurrir en problemas de overfitting o sobreajuste (concepto que presentaremos dentro de poco).



## Modelos no paramétricos: K vecinos más cercanos (KNN)

Un ejemplo sencillo de modelo no paramétrico es el K vecinos más cercanos (KNN). Este modelo simplemente "mira" los K puntos del set de entrenamiento que están más cerca del input de la instancia sobre la cual se quiere realizar la predicción. Este método es un ejemplo de aprendizaje basado en memoria o en instancias.

Formalmente:

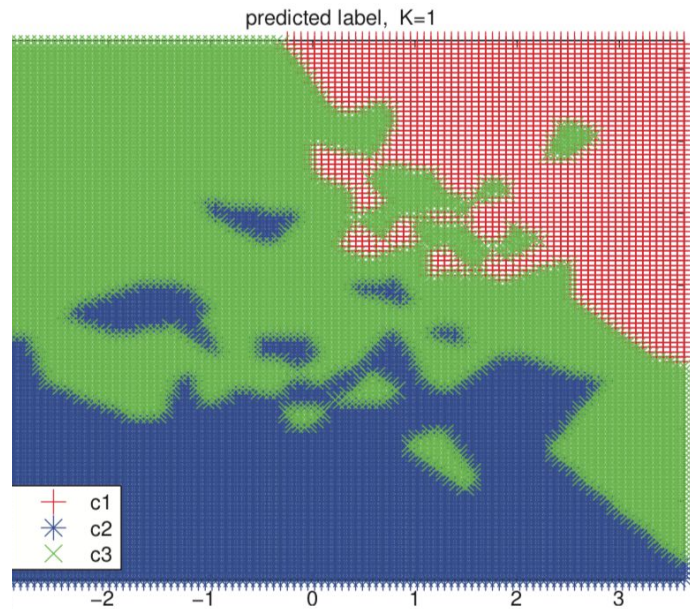
$$p(y = c | \mathbf{x}, \mathcal{D}, K) = \frac{1}{K} \sum_{i \in N_K(\mathbf{x}, \mathcal{D})} \mathbb{I}(y_i = c)$$

$$\mathbb{I}(e) = \begin{cases} 1 & \text{if } e \text{ is true} \\ 0 & \text{if } e \text{ is false} \end{cases}$$

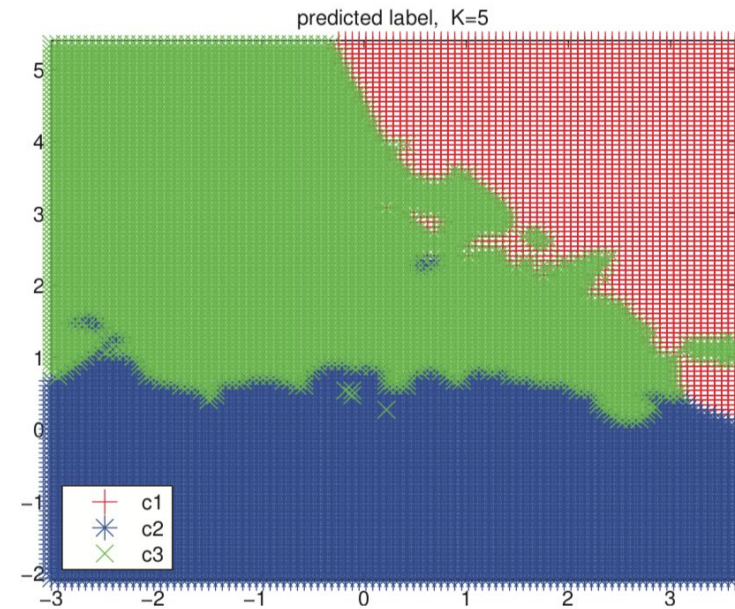
$N_K(\mathbf{x}, \mathcal{D})$  son los índices de los K puntos más cercanos a  $\mathbf{x}$  en el dataset de entrenamiento,  $\mathcal{D}$ .

## Modelos no paramétricos: K vecinos más cercanos (KNN)

La selección de la cantidad de vecinos  $K$  es un parámetro que el algoritmo no aprende de los datos, es necesario predefinirlo. En este sentido, lo denominamos **hiperparámetro**. A diferentes valores de  $K$ , el algoritmo KNN realizará diferentes predicciones. En lo que queda del curso, vamos a darle mucha importancia al modo de optimizar la **selección de hiperparámetros**.



(a)



(b)

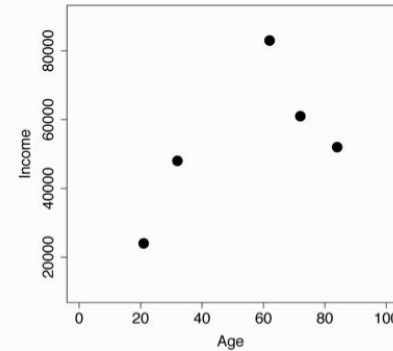
# Underfitting y overfitting

**Underfitting:** el modelo es demasiado simple para capturar la relación entre el input y el output.

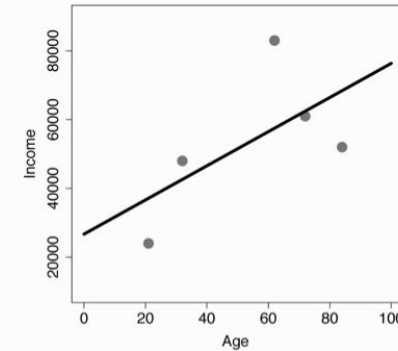
**Overfitting:** el modelo es demasiado complejo y termina ajustándose demasiado bien al dataset de entrenamiento, resultando demasiado sensible al ruido en los datos

The age-income dataset.

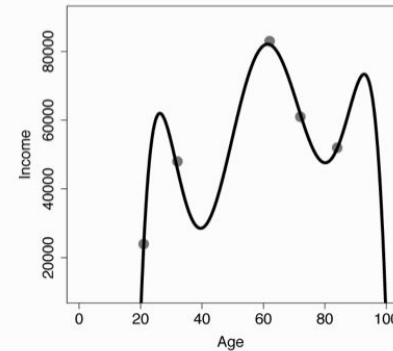
ID	AGE	INCOME
1	21	24,000
2	32	48,000
3	62	83,000
4	72	61,000
5	84	52,000



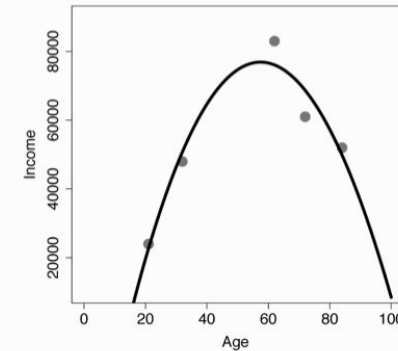
(a) Dataset



(b) Underfitting



(c) Overfitting

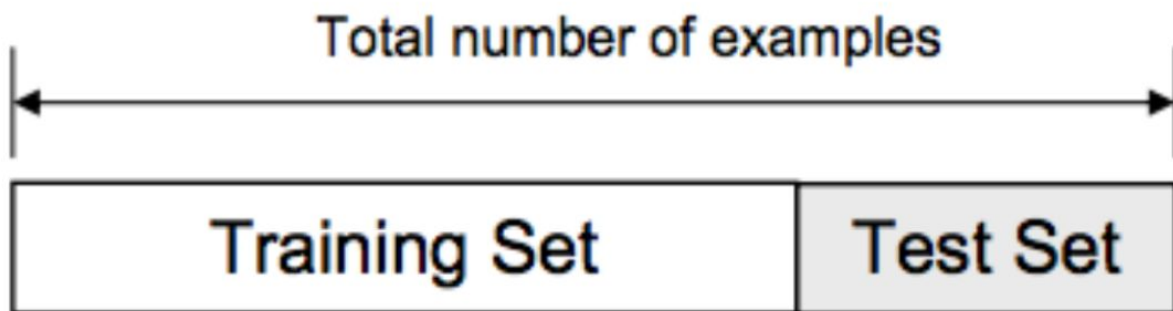


(d) Just right

¿Cómo podemos evaluar si el modelo tiene un problema de overfitting o de underfitting?

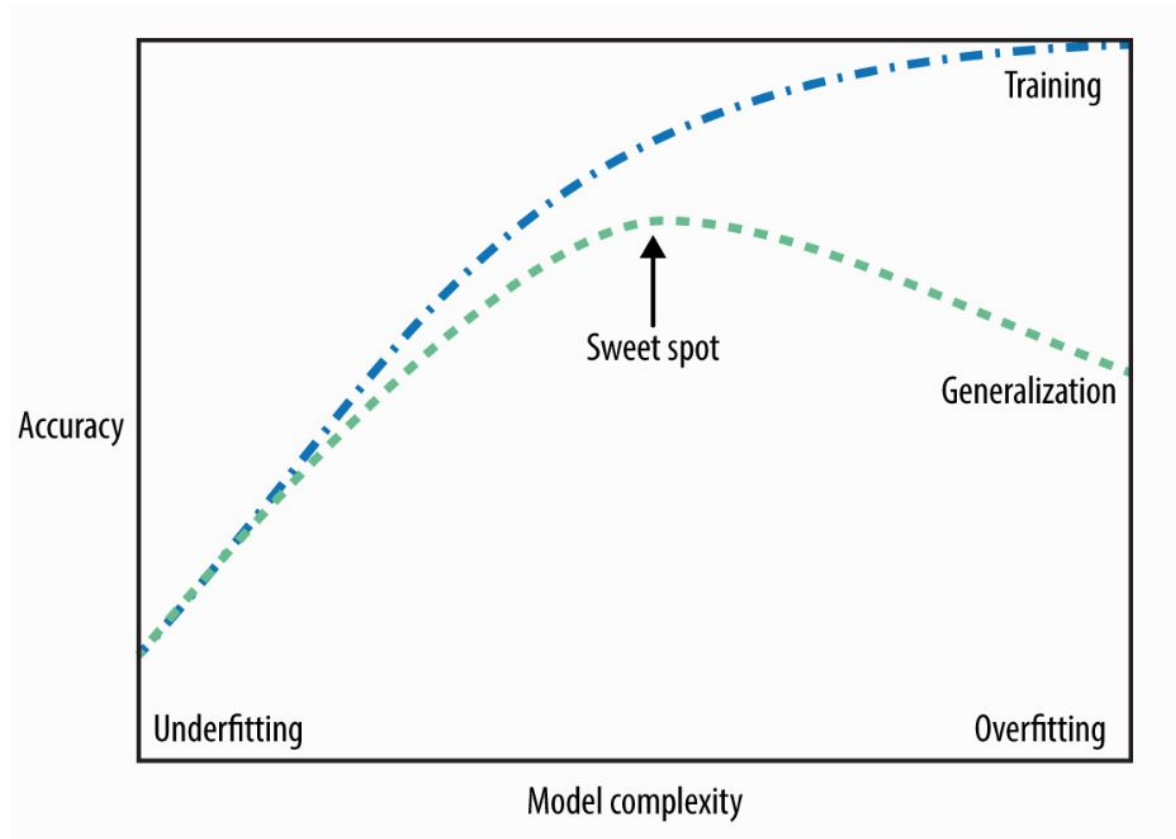
**Separamos los datos en set de entrenamiento y set de testeo.** Vamos a entrenar con los datos del set de entrenamiento y evaluar la performance en el set de testeo.

A lo largo del curso veremos técnicas más sofisticadas, pero todas tienen un elemento en común: evaluar la performance del modelo sobre datos diferentes a los que usamos para entrenar el modelo.



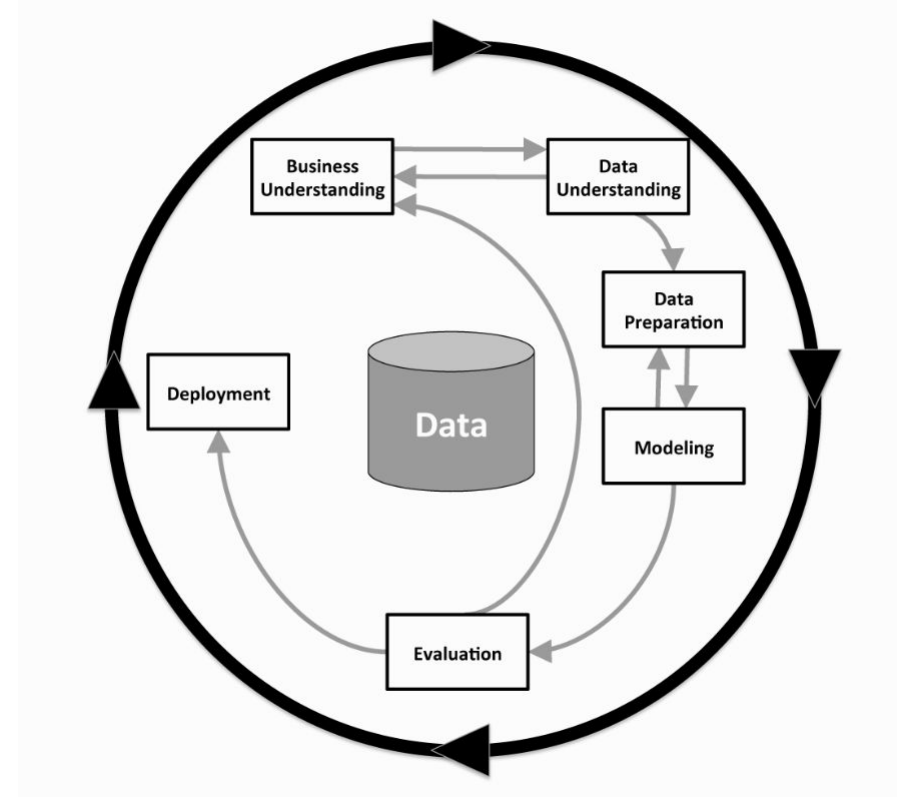
# Underfitting y overfitting

Nuestro objetivo es que el modelo tenga buena capacidad de **generalización**.



## Ciclo de vida de un proyecto: CRISP-DM (Cross-industry standard process for data mining)

- Es uno de los procesos más utilizados para el desarrollo de proyectos de machine learning
- Provee un enfoque estructurado para el planeamiento de un proyecto de machine learning
- Es neutral en cuanto a aplicación, industria y herramientas
- Consiste en 6 etapas presentes en cada proyecto de machine learning. El output de cada etapa determina las tareas que serán desarrolladas en la etapa siguiente.
- El proceso es iterativo, representado que es necesario en un proyecto de machine learning volver a etapas anteriores a partir de lo aprendido.

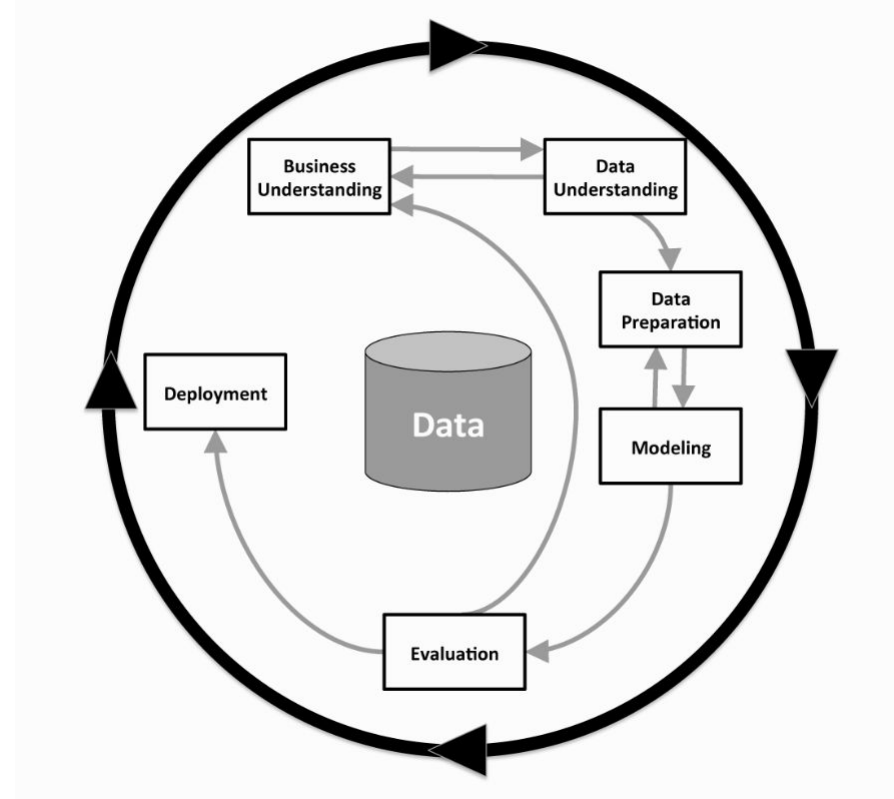


“Fundamentals of Machine Learning for Predictive Data Analytics.” John D. Kelleher, Brian Mac Namee & Aoife D'Arcy(2020)



## CRISP-DM

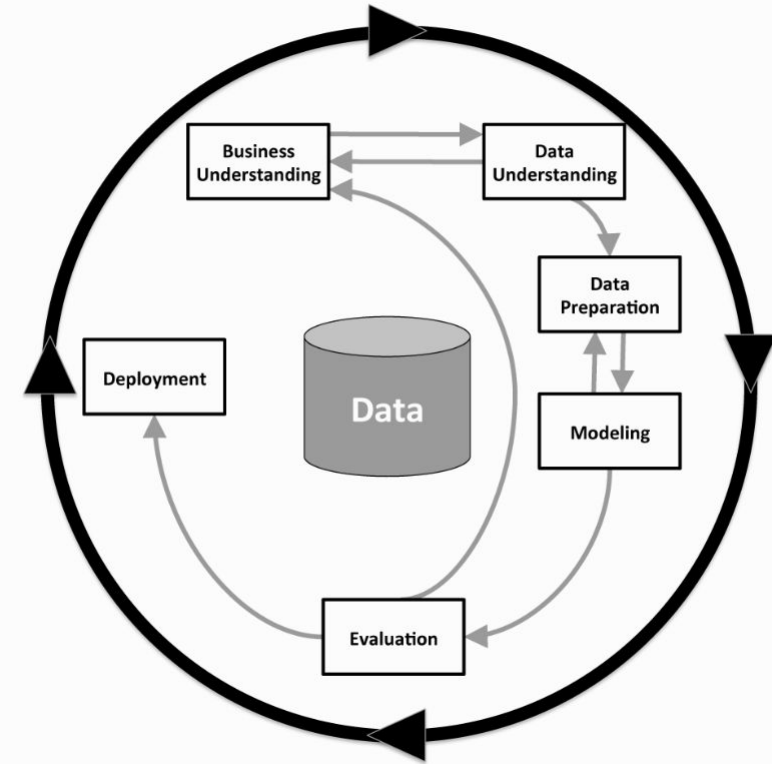
- **Comprensión del negocio:** en la primera etapa, el objetivo del analista es el de entender completamente el problema de negocio, como por ejemplo conseguir más clientes o lograr eficiencias productivas para poder diseñar una solución analítica.
- **Comprensión de los datos:** una vez que se ha decidido la forma en que se utilizará el análisis predictivo de datos para abordar un problema empresarial, es importante que el analista de datos comprenda plenamente las diferentes fuentes de datos disponibles en una organización y los diferentes tipos de datos que contienen estas fuentes.



"Fundamentals of Machine Learning for Predictive Data Analytics." John D. Kelleher, Brian Mac Namee & Aoife D'Arcy(2020)

## CRISP-DM

- **Preparación de los datos:** la construcción de modelos de análisis de datos predictivos requiere tipos específicos de datos, organizados en un tipo específico de estructura conocida como analytics base table (ABT). Esta fase de CRISP-DM incluye todas las actividades necesarias para convertir las fuentes de datos dispares que están disponibles en una organización en una ABT bien formada a partir de la cual se pueden inducir modelos de machine learning.
- **Modelado:** en la fase de modelado del proceso CRISP-DM se produce el trabajo de machine learning. Se utilizan diferentes algoritmos de machine learning para construir una serie de modelos de predicción de los cuales se seleccionará el mejor modelo para su deployment o puesta en producción.

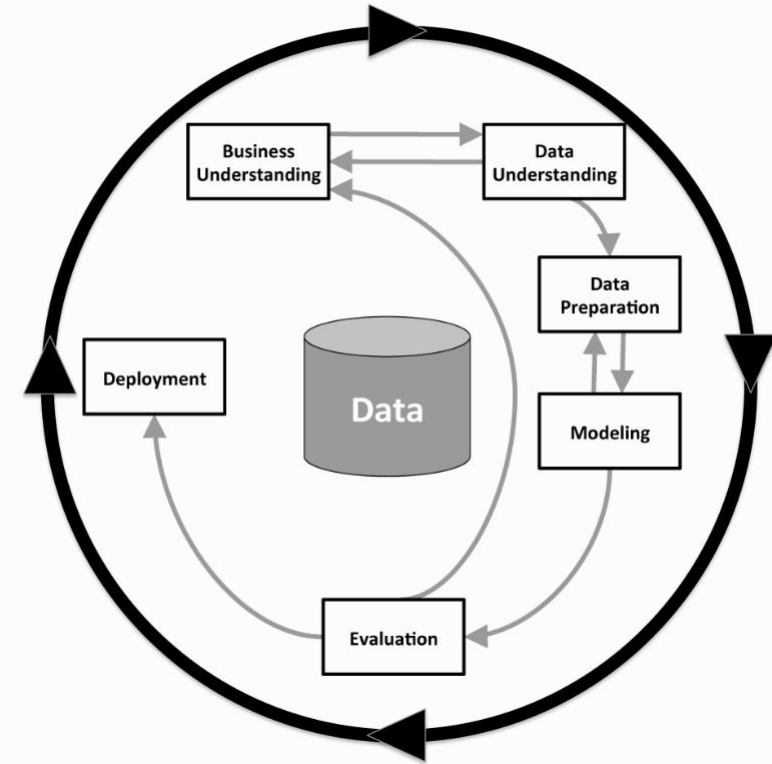


“Fundamentals of Machine Learning for Predictive Data Analytics.” John D. Kelleher, Brian Mac Namee & Aoife D'Arcy(2020)



## CRISP-DM

- **Evaluación:** antes de que los modelos puedan utilizarse en una organización, es importante que se evalúen por completo y se demuestre que son adecuados para su objetivo. Esta fase de CRISP-DM cubre todas las tareas de evaluación necesarias para demostrar que un modelo de predicción será capaz de hacer predicciones precisas después de ser puesto en producción y que no sufre de underfitting u overfitting.
- **Deployment:** los modelos de machine learning se construyen para servir a un propósito dentro de una organización, y la última fase de CRISP-DM cubre todo el trabajo que debe hacerse para integrar con éxito un modelo de machine learning en los procesos dentro de una organización.



“Fundamentals of Machine Learning for Predictive Data Analytics.” John D. Kelleher, Brian Mac Namee & Aoife D'Arcy(2020)

**Scikit-learn** es probablemente la librería más útil para Machine Learning en Python, es de código abierto y es reutilizable en varios contextos, fomentando el uso académico y comercial. Proporciona una gama de algoritmos de aprendizaje supervisados y no supervisados en Python.

Está construida sobre SciPy (Scientific Python) e incluye las siguientes librerías o paquetes: Numpy, Pandas, SciPy, Matplotlib, entre otras.



## Recapitulando:

- Repasamos la definición de machine learning y analizamos sus diferencias con los modelos basados en reglas.
- Repasamos la diferencia entre **aprendizaje supervisado y no supervisado** y una clasificación de los **algoritmos de machine learning**.
- Presentamos la diferencia entre **modelos paramétricos y no paramétricos** y vimos ejemplos de cada caso.
- Presentamos conceptos fundamentales de machine learning como **overfitting vs. underfitting** y el split entre **training y test sets**.
- Presentamos el proceso **CRISP-DM**
- Presentamos a la librería **Scikit-learn**.