

Metodología en Machine Learning

Aprendizaje de Máquina Aplicado

Juan David Martínez Vargas, Ph.D.

jdmartinev@eafit.edu.co

2022

Esteban López

Tomás Olarte

Agenda

- ¿Cómo aprende una computadora?
- Ciclo de vida de un proyecto de ML
- Taxonomía del ML
- Métricas comunes de desempeño
- Sesgo y varianza
- Validación de modelos de ML
- Preprocesamiento de datos



¿Cómo aprende una computadora?

¿Cómo aprende una computadora usando ML?

Un algoritmo de ML aprende mediante un **modelo** o función de hipótesis (a menudo denotado h) cuyos **parámetros** o reglas se ajustan a los datos.

El modelo h es simplemente una función que recibe unas **características** y devuelve una **predicción**.

$h(\text{🍏}) = \text{manzana}$

$h(\text{🍅}) = \text{tomate}$

$h(\text{🐮}) = \text{vaca}$

Parámetros e hiperparámetros de un modelo de ML

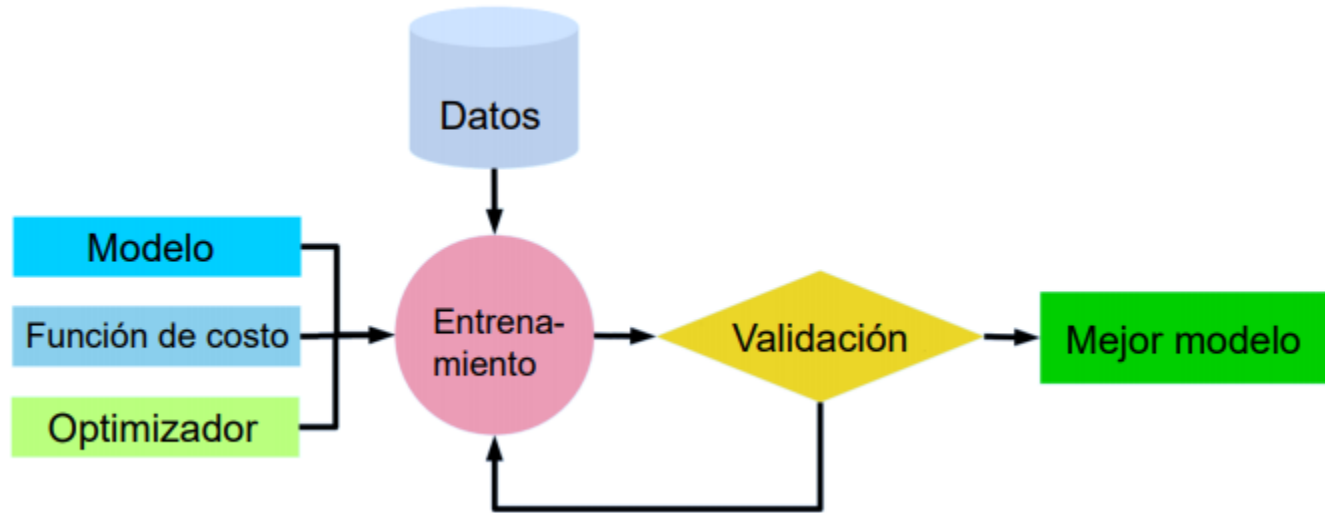
Parámetros:

- Se establecen mediante el entrenamiento
- Se deben guardar como parte del modelo
- **Ejemplos:** Los pesos de una regresión lineal o los puntos de corte de un árbol de decisión

Hiperparámetros:

- Se establecen antes de entrenar (a menudo mediante un proceso de validación)
- Por lo general no es necesario guardarlos para reproducir el modelo
- **Ejemplos:** El parámetro de regularización de una regresión lineal o la profundidad máxima de un árbol de decisión

¿Cómo aprende una computadora usando ML?



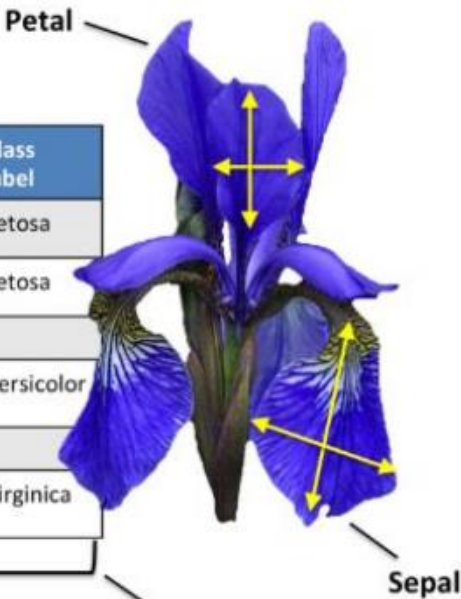
¿Cómo Son los Datos que Ingiera un Algoritmo de ML?

Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)




Las **características** se suelen denotar con **X** y la variable a predecir se suele denotar con **y**.

La variable a predecir a menudo se denomina **etiqueta**.

Aplicaciones del Machine Learning

- Detectar spam
- Detectar fraudes
- Recomendaciones de productos y servicios
- Diagnósticos médicos
- Predicción de fuga de clientes
- Segmentación de clientes
- Predicción de demanda de energía
- Clasificación de imágenes
- Detección de objetos en imágenes
- Clasificación de texto
- Traducción
- Obtención de resúmenes de texto
- Reconocimiento de voz
- Generación de texto e imágenes
- Descripción de imágenes

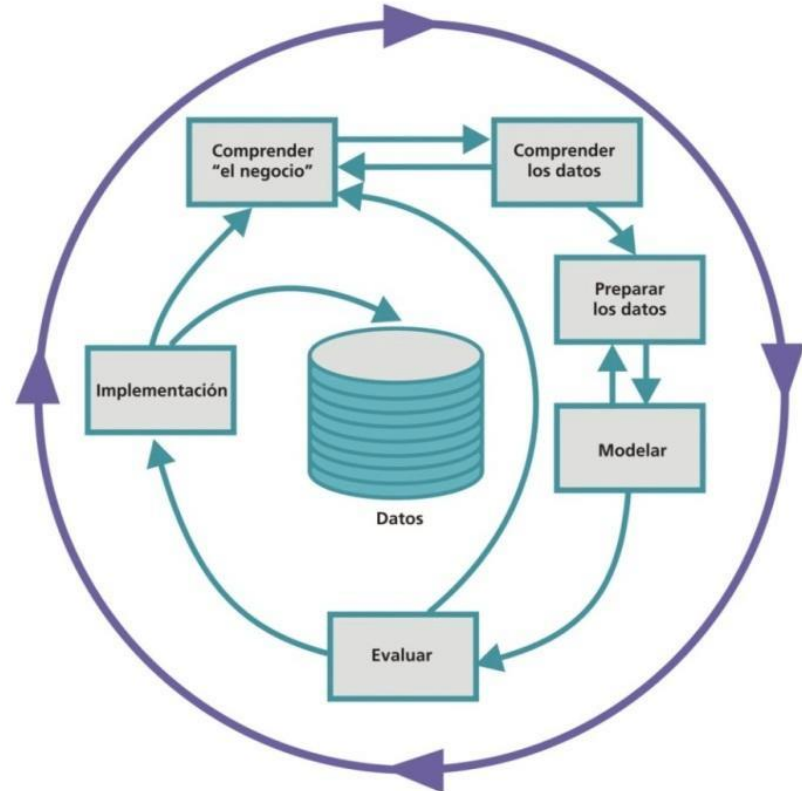


Ciclo de vida de un proyecto de Machine Learning

Pasos para entrenar un modelo de ML

- 1 **Recolectar los datos.** Podemos recolectar los datos desde muchas fuentes, podemos por ejemplo extraer los datos de un sitio web o obtener los datos utilizando una API, desde una base de datos, desde otros dispositivos...
- 2 **Preprocesar los datos.** Una vez que tenemos los datos, tenemos que asegurarnos que tiene el formato correcto para nutrir nuestro algoritmo de aprendizaje.
- 3 **Explorar los datos.** Una vez que ya tenemos los datos y están con el formato correcto, podemos realizar un pre análisis para corregir los casos de valores faltantes o intentar encontrar a simple vista algún patrón en los mismos que nos facilite la construcción del modelo.
- 4 **Entrenar el Algoritmo.** En esta etapa nutrimos al o a los algoritmos de aprendizaje con los datos que venimos procesando en las etapas anteriores.
- 5 **Evaluar el Algoritmo.** En esta etapa ponemos a prueba la información o conocimiento que el algoritmo obtuvo del entrenamiento del paso anterior. Evaluamos la precisión del algoritmo en sus predicciones y si no estamos conformes con su rendimiento, podemos volver a la etapa anterior y continuar entrenando.
- 6 **Utilizar el modelo.** Medimos su rendimiento, lo que tal vez nos obligue a revisar todos los pasos anteriores.

Ciclo de Vida de un Proyecto de Machine Learning



Cross Industry Standard Process
for Data Mining (CRISP-DM)



Taxonomía del Machine Learning

Clasificación de los Datos

ESTRUCTURADOS

Datos que tienen un modelo definido o provienen de un campo determinado en un registro



precios de acciones - base de datos de compras - rastreo web

NO ESTRUCTURADOS

Datos que no tienen un modelo predefinido o no están organizados de alguna manera



fotografía - documentos de texto - video

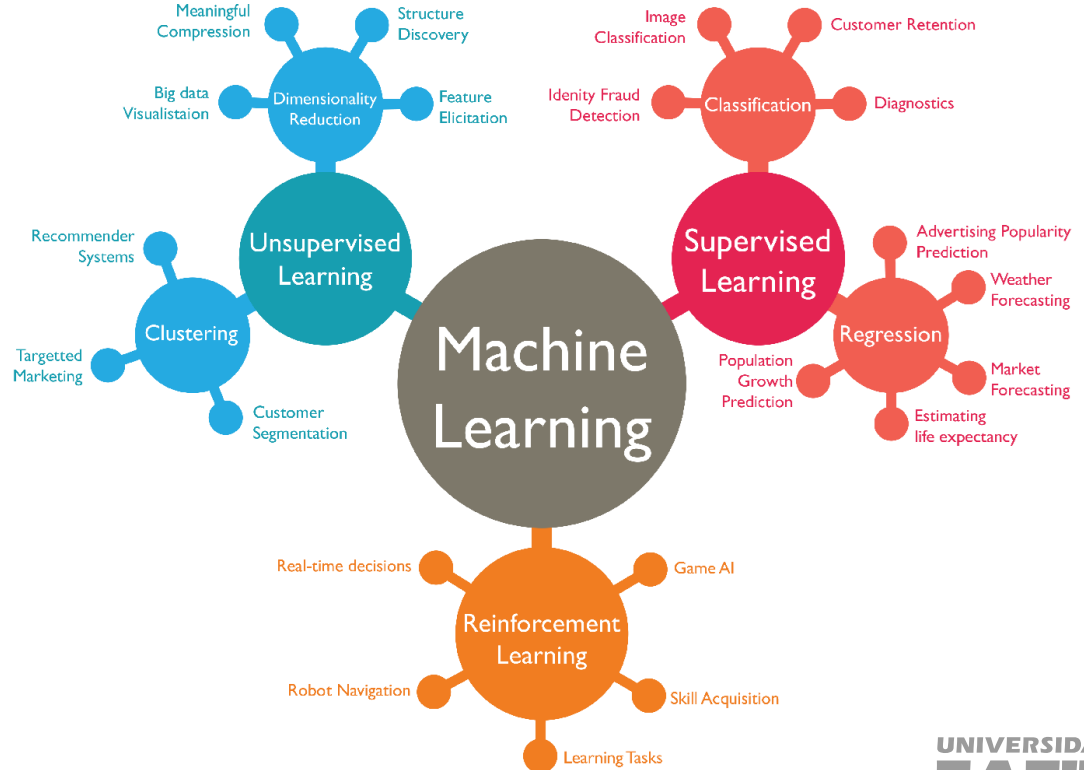
Clasificación de Características Estructuradas

- **Numeric:** valores que permiten operaciones aritméticas (precios, edad, magnitudes físicas)
- **Interval:** Valores que permiten ordenación o sustracción pero no más operaciones aritméticas (fechas)
- **Ordinal:** valores que permiten un orden pero no operaciones aritméticas (por ejemplo identificadores, tallas M, L, XL...)
- **Categorical:** Es un conjunto finito de valores no ordenable y que no permite operaciones aritméticas (tipos de producto, países, idiomas etc)
- **Binary:** Solo admite dos posibles valores (género)
- **Textual:** Texto en formato libre de diferentes extensiones (nombre, direcciones etc)

Tipos de Aprendizaje en ML

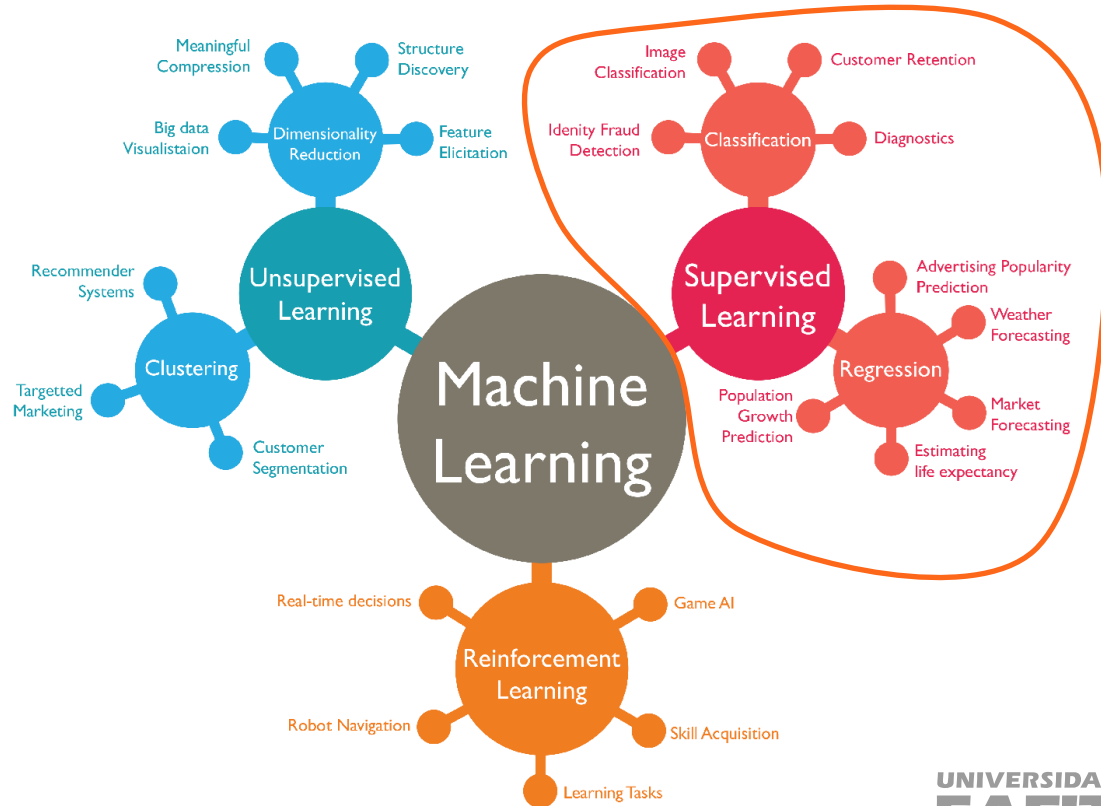
Existen tres tipos principales de aprendizaje en ML:

- Aprendizaje supervisado
- Aprendizaje no supervisado
- Aprendizaje por refuerzo

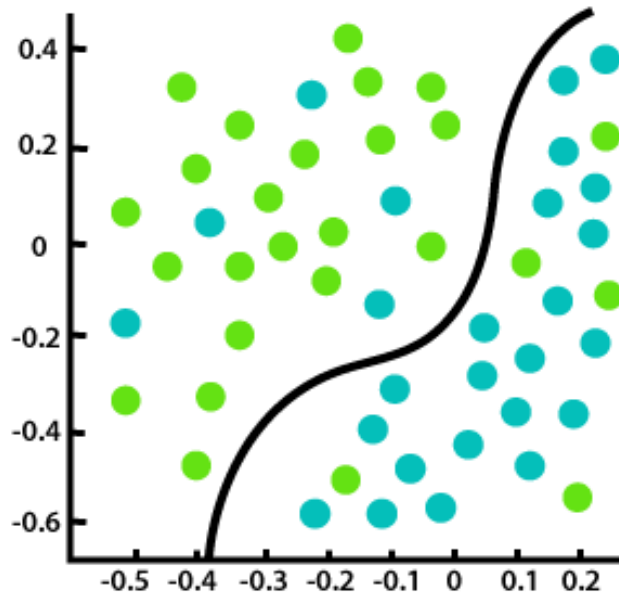


Tipos de Aprendizaje en ML

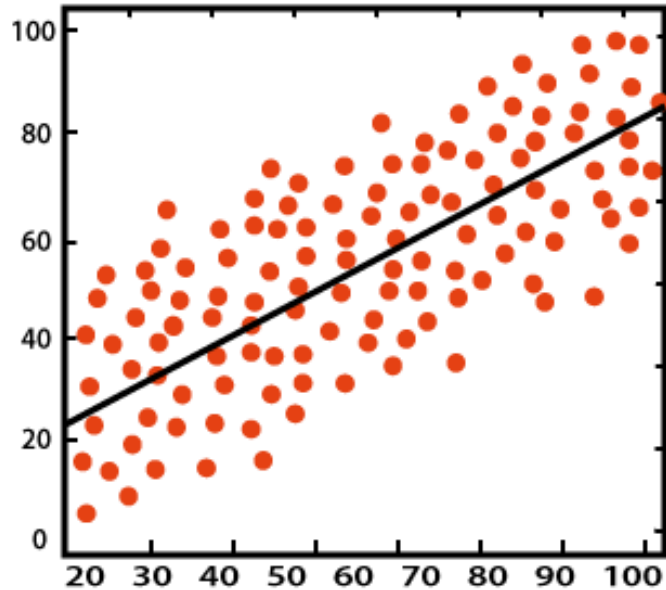
Por ahora nos
concentraremos en
el aprendizaje
supervisado



Tipos de Aprendizaje Supervisado



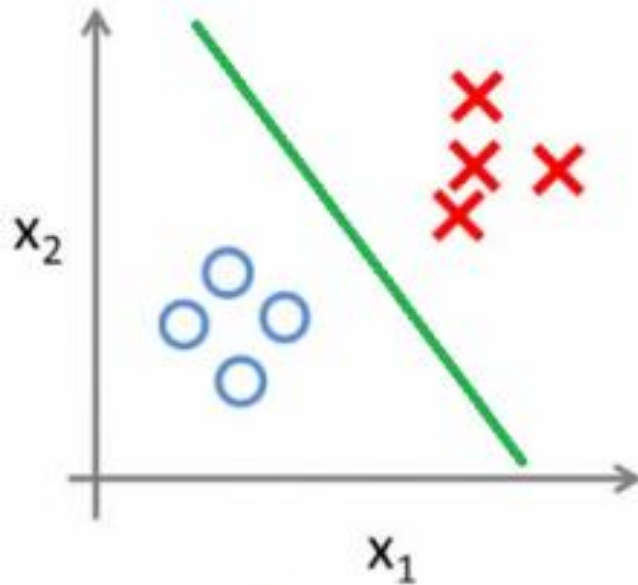
Classification



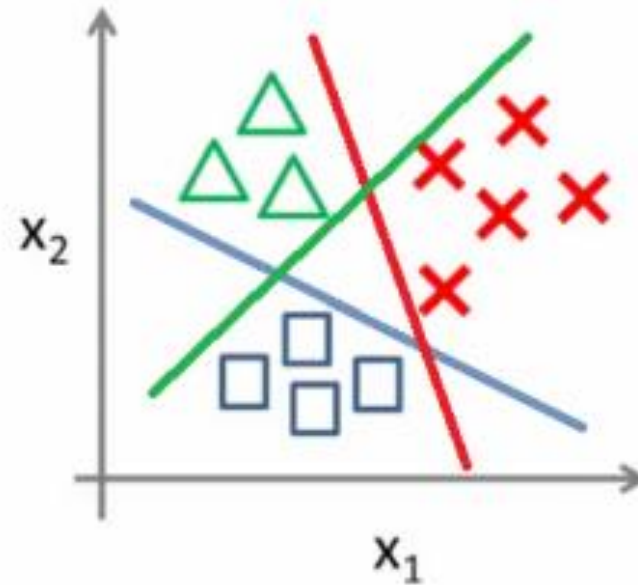
Regression

Tipos de Clasificación

Binary classification:



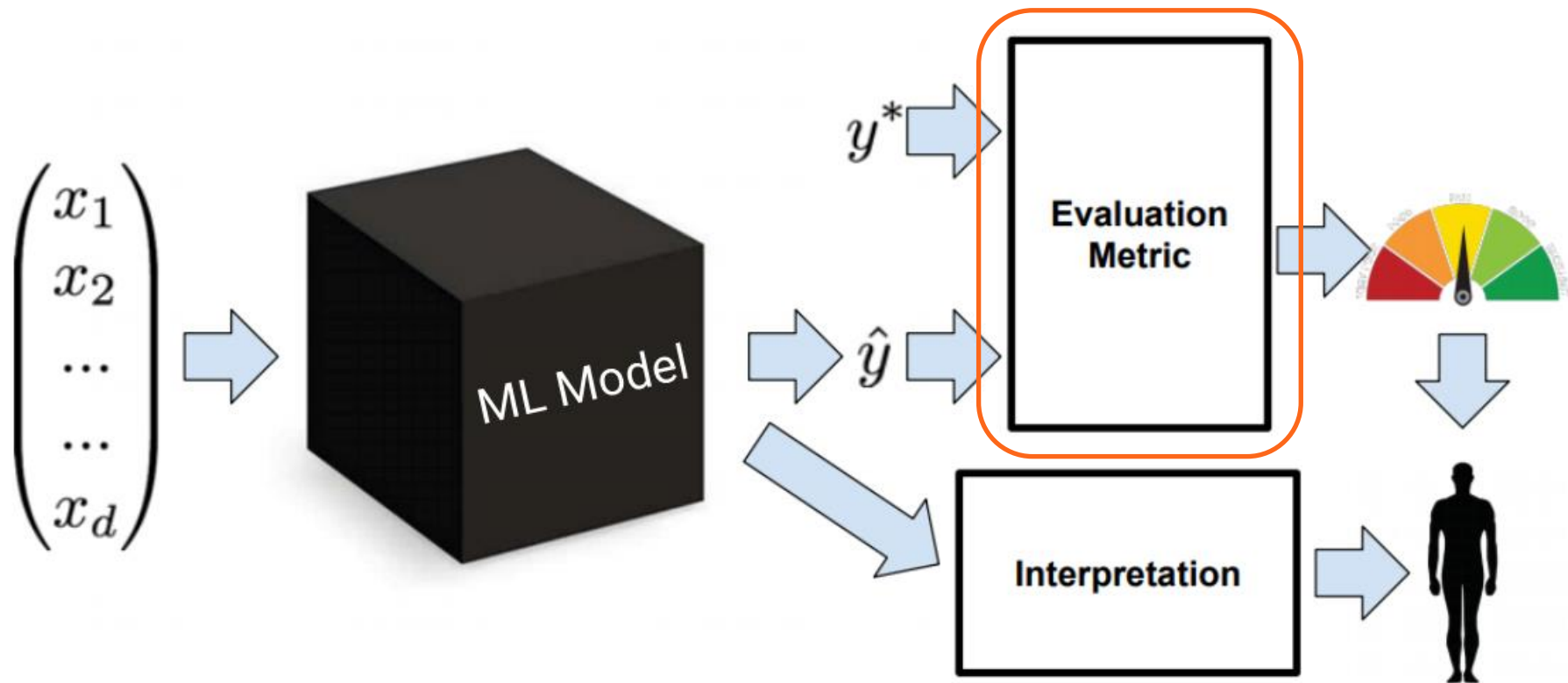
Multi-class classification:





Métricas de desempeño en aprendizaje supervisado

Evaluación de un Modelo de ML



Métricas Comunes en Aprendizaje Supervisado

Métricas de regresión:

- Error cuadrático medio (MSE)
- Error absoluto medio (MAE)
- Raíz cuadrada del error cuadrático medio (RMSE)
- Error absoluto porcentual medio (MAPE)

Métricas de clasificación:

- Accuracy
- Precision
- Recall
- F1-score

Métricas Comunes de Regresión

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

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

\hat{y} – predicted value of y

\bar{y} – mean value of y

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Métricas Comunes de Clasificación

		predicted condition		
		prediction positive	prediction negative	
true condition	total population			Sensitivity
	condition positive	True Positive (TP)	False Negative (FN) (Type II error)	Recall = $\frac{\sum TP}{\sum \text{condition positive}}$
	condition negative	False Positive (FP) (Type I error)	True Negative (TN)	Specificity = $\frac{\sum TN}{\sum \text{condition negative}}$
		Precision= $\frac{\sum TP}{\sum \text{prediction positive}}$		F1 Score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
		Accuracy = $\frac{\sum TP + \sum TN}{\sum \text{total population}}$		

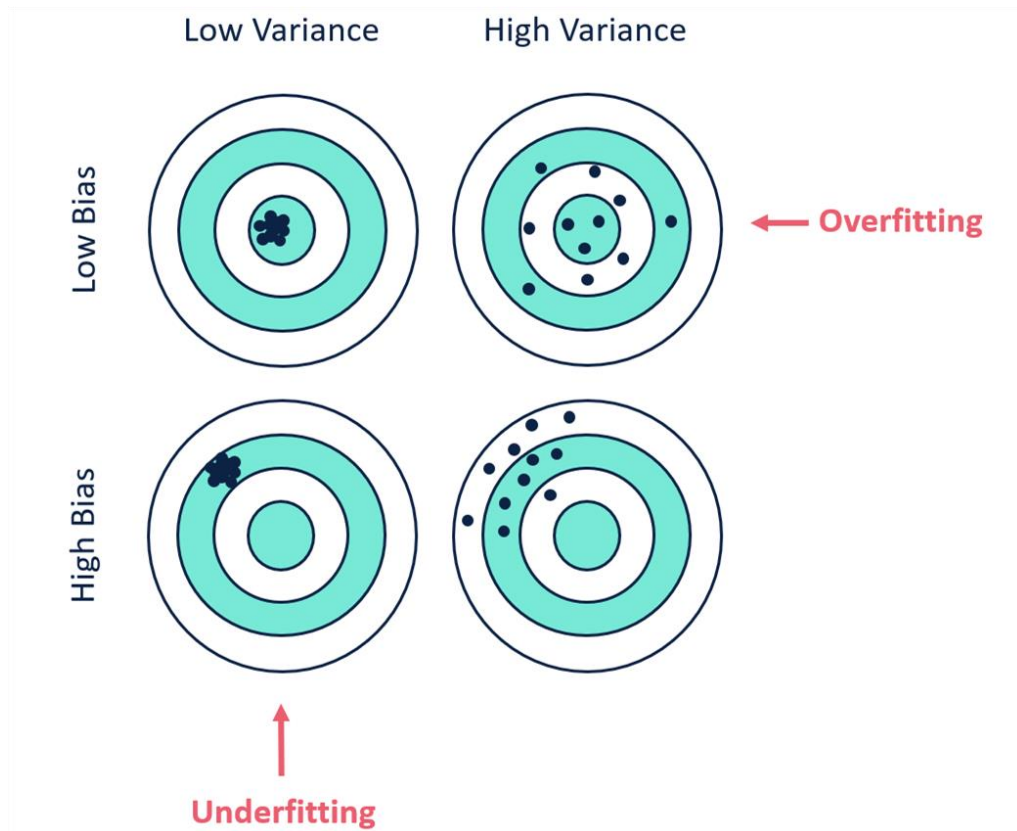


Sesgo y Varianza en Machine Learning

Si no hay errores de código y los datos están limpios y tienen sentido, hay dos fuentes principales de error en los modelos de ML:

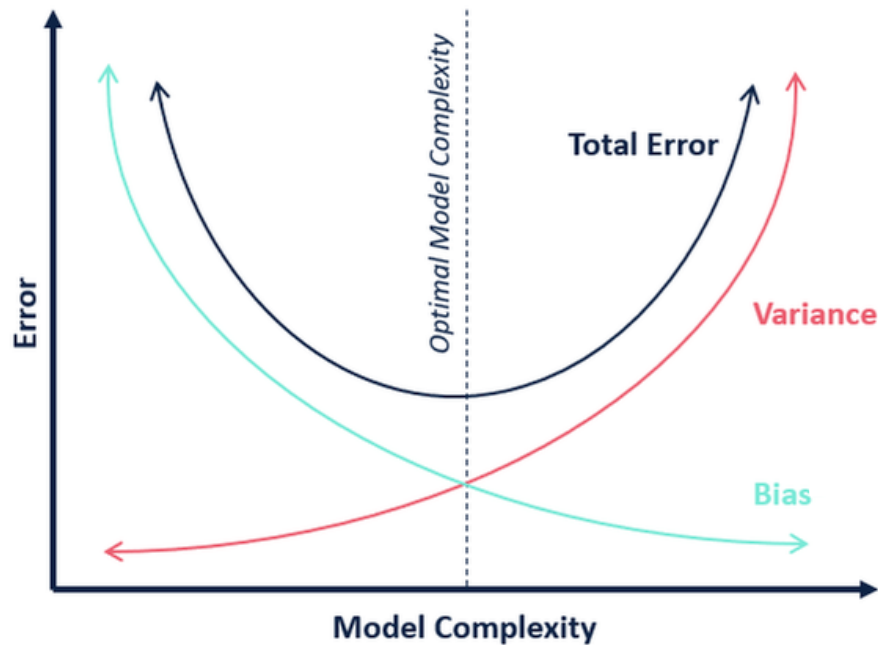
- Sesgo (bias)
- Varianza (variance)

Sesgo y Varianza

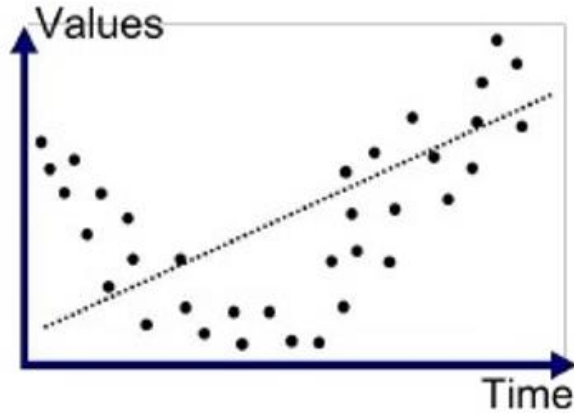


Sesgo y Varianza

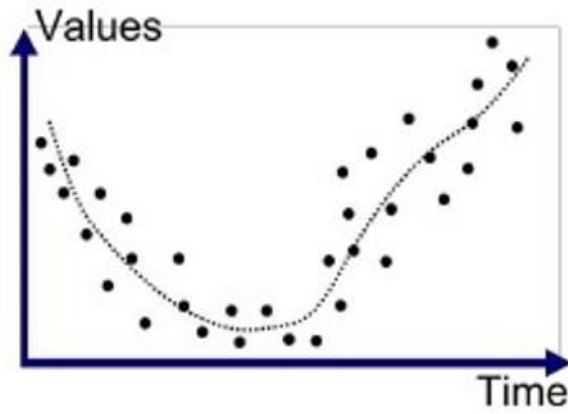
Por lo general, asociamos el sesgo alto con modelos muy simples que **subajustan (underfit)** los datos y asociamos la varianza alta con modelos muy complejos que **sobreajustan (overfit)** los datos.



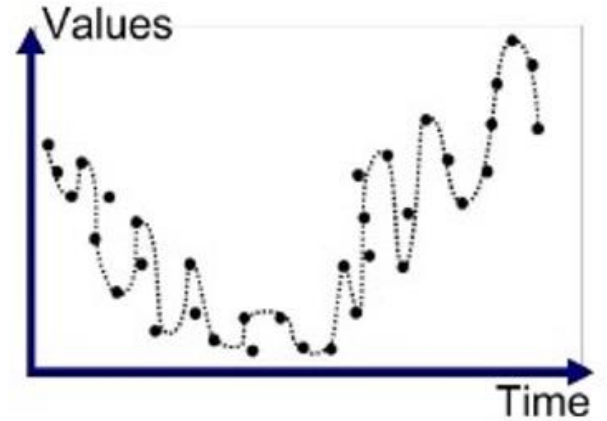
Subajuste y Sobreajuste



Underfitted

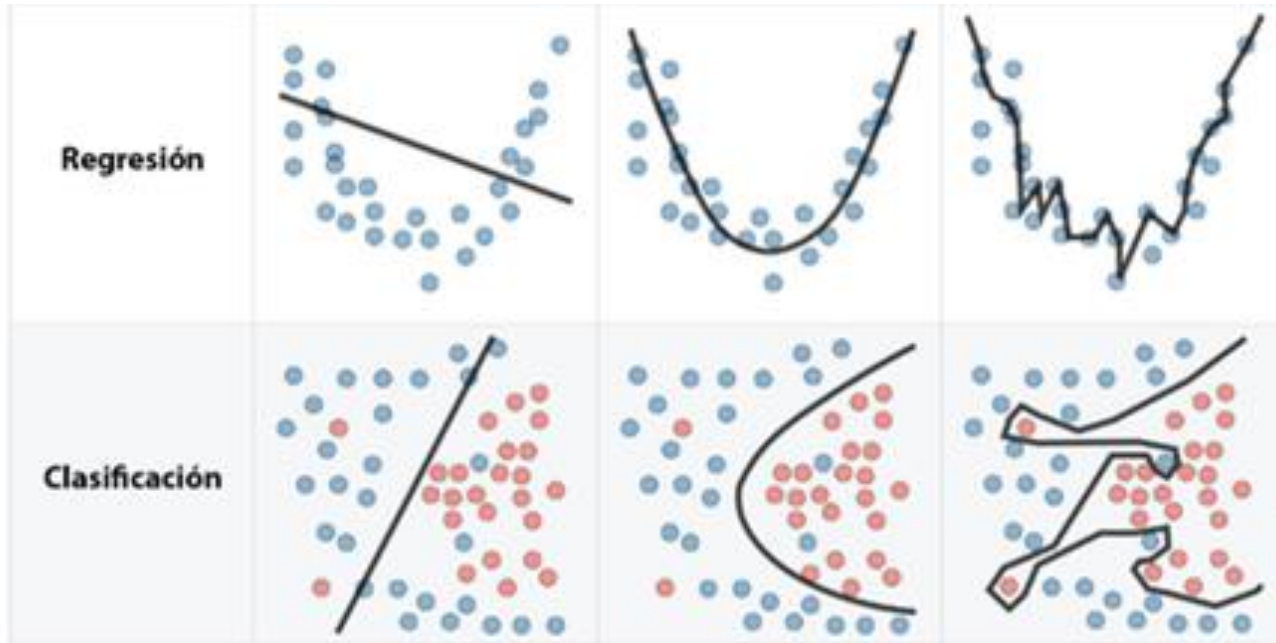



Good Fit/Robust



Overfitted

Subajuste y Sobreajuste





Evaluación y Desarrollo de Modelos de Machine Learning

Evaluación de un Modelo de ML

¿Cómo podemos saber si un modelo de Machine Learning generaliza bien?

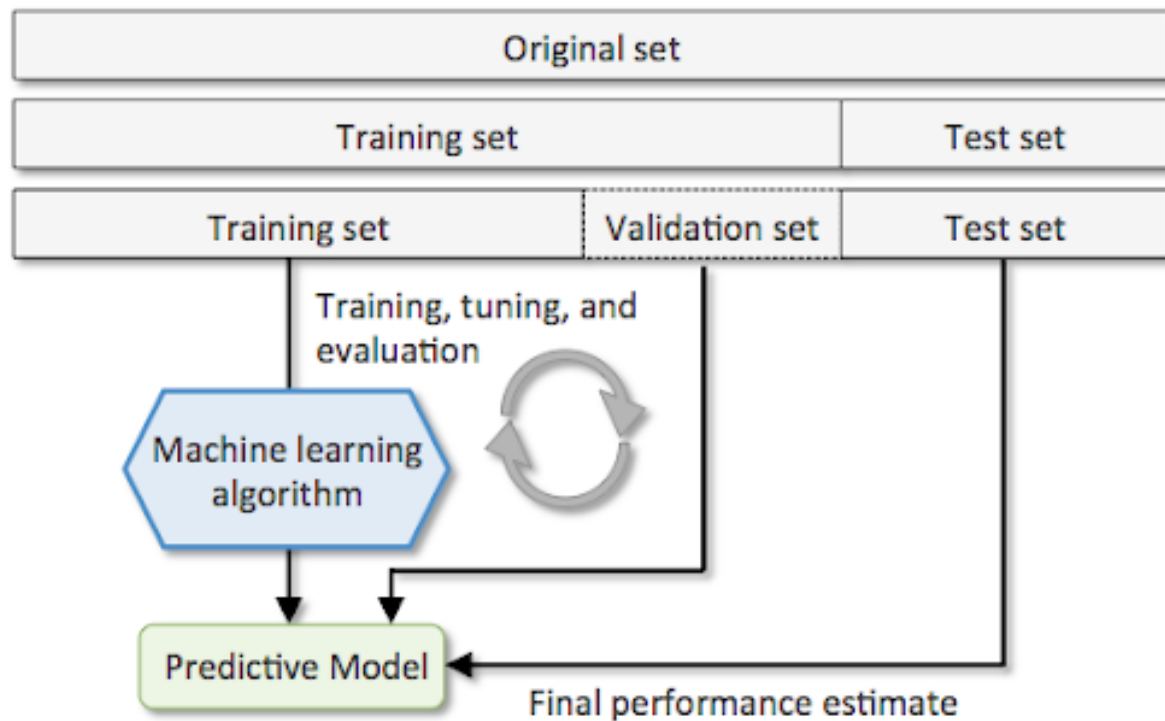
Respuesta: Se debe evaluar el modelo en un conjunto de datos con el que no se haya entrenado (**conjunto de prueba**).

Desarrollo de un Modelo de ML

¿Cómo podemos obtener un mejor modelo si el actual no generaliza bien?

Respuesta: Se selecciona el tipo de modelo y sus hiperparámetros (**desarrollo del modelo**) de tal forma que su desempeño en un **conjunto de validación** sea bueno.

Evaluación y Desarrollo de un Modelo de ML



Conjuntos de Entrenamiento, Validación y Prueba

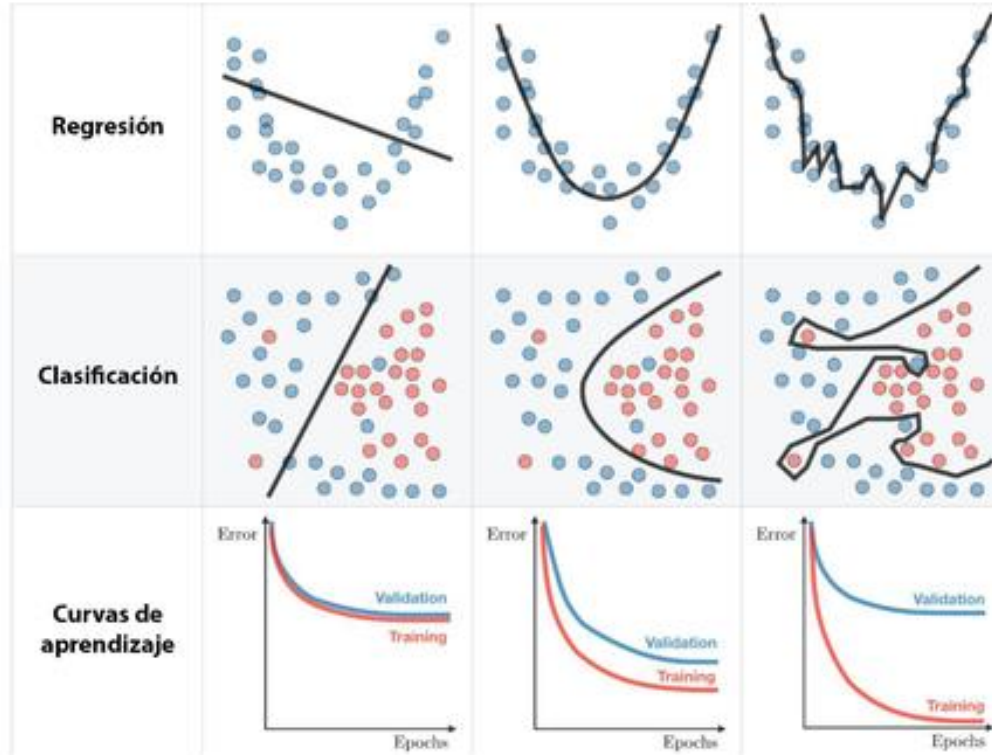
- **Conjunto de entrenamiento:** se utiliza para hallar los parámetros o reglas del modelo.
- **Conjunto de validación o desarrollo:** no debe usarse en el entrenamiento. Se utiliza para hallar los hiperparámetros del modelo
- **Conjunto de prueba:** no debe usarse en el entrenamiento ni en el desarrollo para poder saber cómo se desempeña el modelo frente a datos que no intervinieron en la selección de parámetros ni hiperparámetros.

Conjuntos de Entrenamiento, Validación y Prueba

Los conjuntos de entrenamiento, validación y prueba se deben escoger aleatoriamente (y posiblemente estratificados por la etiqueta). Algunas divisiones comunes son:

- 70% entrenamiento, 15% validación, 15% prueba
- 80% entrenamiento, 10% validación, 10% prueba
- 60% entrenamiento, 20% validación, 20% prueba

Curvas de Aprendizaje



Conjuntos de Entrenamiento, Validación y Prueba

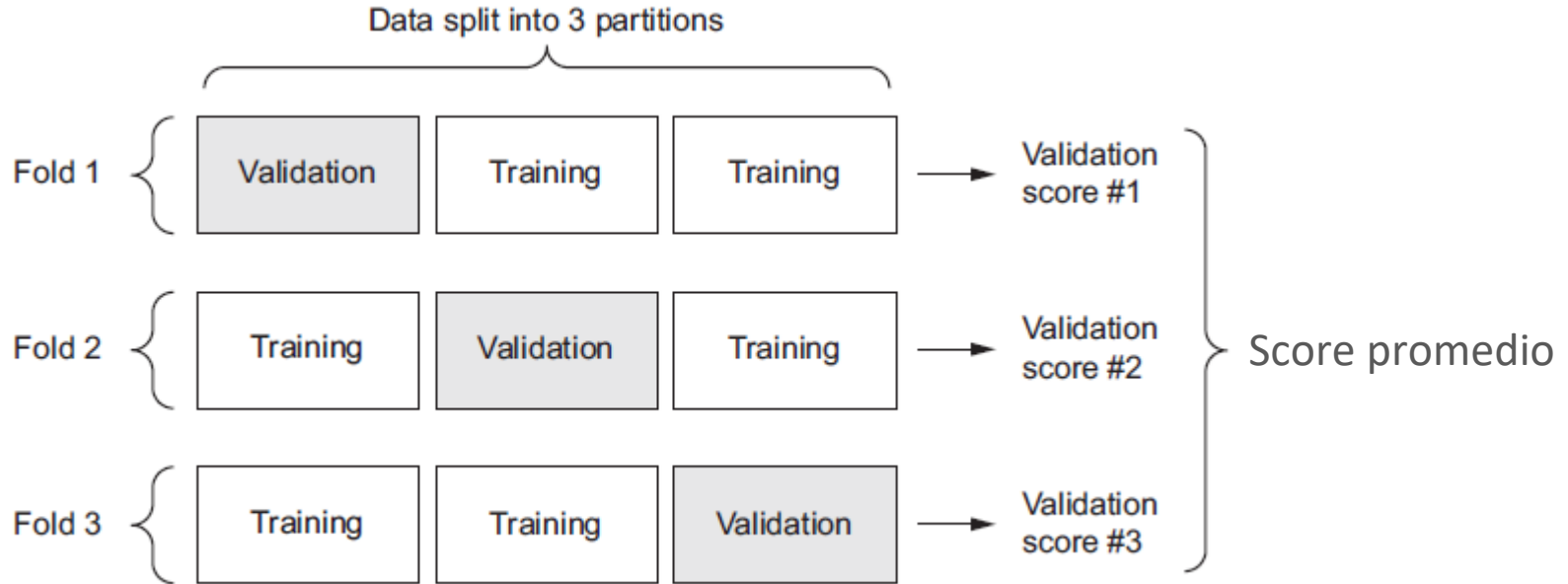
En algunos casos, por ser muy simples, por contar con pocos datos o por ser de carácter académico, se prescinde del conjunto de validación. En estos casos:

- No se hace desarrollo del modelo.
- Se hace desarrollo usando el conjunto de prueba y se acepta la incertidumbre sobre la capacidad del modelo para generalizar.

Validación Cruzada

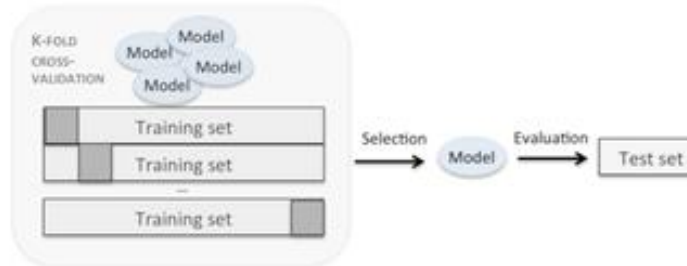
En otros casos, cuando el modelo empleado se puede entrenar rápidamente, es posible usar **validación cruzada** para seleccionar los hiperparámetros del modelo.

Validación Cruzada




K-fold cross-validation

Evaluación y Desarrollo de un Modelo de ML



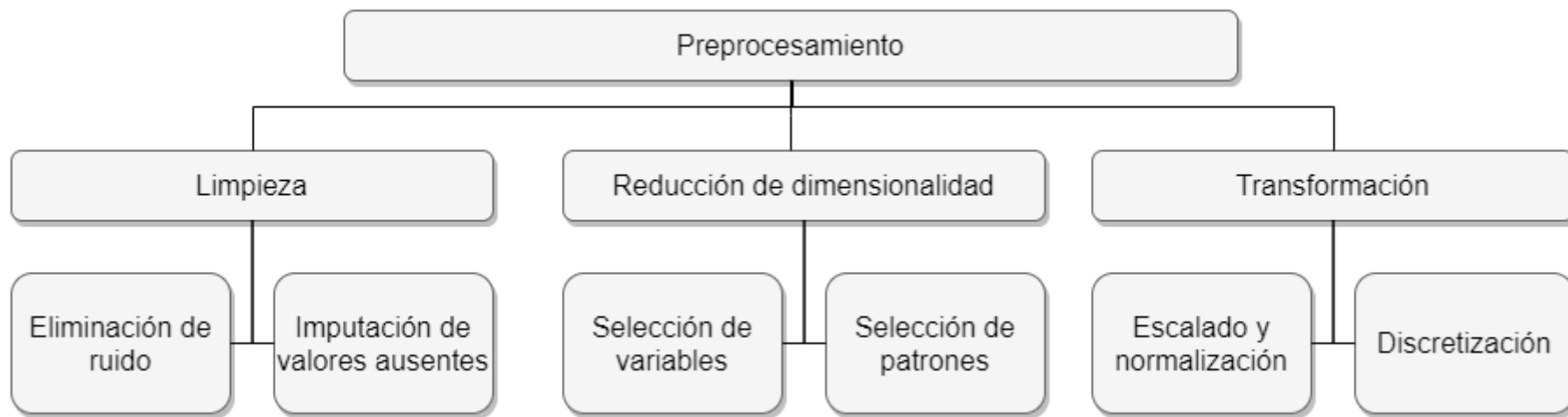
Evaluación y Desarrollo de un Modelo de ML

En cualquier caso, si se aplicó alguna forma de validación para seleccionar los hiperparámetros del modelo y ya no se va a hacer más desarrollo, es común reentrenar el modelo usando los datos de entrenamiento y de validación para mejorar un poco el desempeño final.



Preparando los datos para aplicar Machine Learning

Tareas comunes de procesamiento



Procesamiento de Variables Categóricas

Las variables categóricas deben ser apropiadamente codificadas. Para la mayoría de modelos, a estas variables se les debe aplicar **one-hot encoding**. Por otro lado, para los modelos basados en árboles de decisión, estas variables pueden codificarse como enteros sucesivos (**ordinal encoding**).

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50



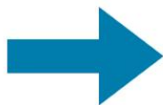
One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Procesamiento de Variables Categóricas

Otro ejemplo:

Color
Red
Red
Yellow
Green
Yellow




Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

Procesamiento de Variables Binarias

Las variables binarias deben ser codificadas con ceros y unos (sin generar columnas extra).

Condición	Enfermo	Sano	Sano	Enfermo	Enfermo	Sano	Enfermo
	1	0	0	1	1	0	1



Condición							
	1	0	0	1	1	0	1

0

Enfermo							
	1	0	0	1	1	0	1



Ejemplo de preprocesamiento de datos



GRACIAS

UNIVERSIDAD
EAFIT[®]