



# INTRODUCCIÓN PRÁCTICA AL MACHINE LEARNING CON PYTHON

---

**Andrés Merino T.  
Noviembre 2025**

## OBJETIVO Y AGENDA

---

## OBJETIVO Y AGENDA

**Objetivo:** comprender el flujo mínimo para aplicar Machine Learning en Python y reconocer decisiones clave (tipo de problema, datos, validación y métricas).

## Agenda

- ¿Qué es Machine Learning y cuándo usarlo?
- Supervisado vs no supervisado
- Métricas y validación
- Flujo de trabajo en ML
- Demo práctica en Python

## ¿QUÉ ES MACHINE LEARNING?

---

## ¿QUÉ ES MACHINE LEARNING?

- El Aprendizaje Automático (Machine Learning) es una subdisciplina de la Inteligencia Artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las máquinas aprender a partir de datos sin ser explícitamente programadas para cada tarea específica.
- Tom Mitchell (1997): Se dice que un programa de computadora aprende de la experiencia  $E$  con respecto a alguna clase de tareas  $T$  y medida de desempeño  $P$ , si su desempeño en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ .

## EJEMPLOS TÍPICOS DE ML

### Clasificación

- spam / no spam
- fraude / no fraude
- aprobación / reprobación

### Regresión

- demanda
- precio
- tiempo estimado

### No supervisado

- segmentación (clustering), reducción de dimensión, detección de anomalías

## ¿CUÁNDO USAR (Y CUÁNDO NO USAR) ML?

### Usar ML cuando:

- el patrón es complejo y reglas manuales no escalan,
- hay datos suficientes y representativos,
- el objetivo y el costo del error están claros.

### Evitar ML si:

- una regla simple resuelve la mayor parte del caso,
- los datos son escasos/inestables o la etiqueta es poco confiable,
- no existe plan de evaluación y monitoreo.

## CONCEPTOS MÍNIMOS (PARA NO PERDERSE)

- **Feature (variable):** columna de entrada ( $X$ ).
- **Etiqueta (target):** lo que se quiere predecir ( $y$ ).
- **Modelo:** algoritmo + parámetros aprendidos.
- **Entrenar:** ajustar el modelo con datos de entrenamiento.
- **Predecir:** estimar  $y$  para datos nuevos.
- **Generalización:** desempeño en datos no vistos.

## **SUPERVISADO VS NO SUPERVISADO**

---

## SUPERVISADO VS NO SUPERVISADO

### Aprendizaje supervisado

- Datos:  $(X, y)$
- Objetivo: predecir  $y$
- Tareas: clasificación, regresión

### Aprendizaje no supervisado

- Datos:  $X$
- Objetivo: descubrir estructura
- Tareas: clustering, PCA, anomalías

**Pregunta guía:** ¿tengo una etiqueta confiable para aprender?

# SUPERVISADO: CLASIFICACIÓN VS REGRESIÓN

## Clasificación

- salida discreta (clases)
- ejemplo: *riesgo alto/medio/bajo*

**Salida:**  $y \in \{0, 1, \dots\}$

## Regresión

- salida continua
- ejemplo: *monto esperado*

**Salida:**  $y \in \mathbb{R}$

## SUPERVISADO: IDEA MATEMÁTICA

- Disponemos de un conjunto de datos etiquetados:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n,$$

donde  $x_i$  son los datos de entrada (variables, features) y  $y_i$  la etiqueta o salida observada.

- Suponemos que existe una función (desconocida)  $f$  tal que:

$$y = f(x).$$

- El objetivo del aprendizaje supervisado es encontrar una función aproximante  $\tilde{f}$  tal que:

$$\tilde{f}(x) \approx y.$$

## MÉTRICAS Y VALIDACIÓN

---

## MÉTRICAS Y VALIDACIÓN

- Una vez obtenida la función aproximante  $\tilde{f}$ , surge la pregunta:  
¿Qué tan bien aproxima  $\tilde{f}(x)$  a  $y$ ?
- Evaluamos el desempeño mediante una **métrica de error** en datos observados:
  - Clasificación: accuracy, precision, recall, F1
  - Regresión: MAE, MSE, RMSE

- **Sobreajuste (overfitting):**

- $\hat{f}$  se ajusta demasiado a los datos.
- Aprende el ruido en lugar del patrón general.
- Resultado: muy buen desempeño en datos conocidos, mal desempeño en datos nuevos.

- **Motivación del split train/test:**

- Dividimos el conjunto de datos conocidos en dos grupos: train/test.
- Entrenamos  $\hat{f}$  con **train**.
- Evaluamos qué tan bien generaliza usando **test** (datos conocidos pero no vistos).
- El conjunto test actúa como un “examen final” de la función.

## FLUJO DE TRABAJO EN ML

---

## FLUJO DE TRABAJO EN MACHINE LEARNING

### 1. Entender el problema (contexto/negocio)

- ¿Qué decisión se quiere apoyar? ¿Cuál es el objetivo?
- ¿Qué significa un error? ¿Cuál es su costo?

### 2. Entender los datos

- ¿Qué variables existen? ¿Cómo se generan?
- Revisar calidad: nulos, ruido, sesgos, cobertura.

### 3. Definir el problema de ML

- Tipo: clasificación, regresión o no supervisado.
- Variable objetivo, unidad de análisis y horizonte.

## FLUJO DE TRABAJO EN MACHINE LEARNING

### 4. Preparar los datos

- Limpieza, selección/creación de variables, escalado/categóricas.
- Separar en **train** y **test**.

### 5. Entrenar el modelo

- Empezar con un baseline simple.
- Ajustar el modelo con datos de entrenamiento.

### 6. Medir desempeño

- Evaluar en entrenamiento (diagnóstico) y en test (generalización).
- Usar métricas adecuadas al problema.

## FLUJO DE TRABAJO EN MACHINE LEARNING

### 7. Iterar y mejorar

- Mejorar datos, variables, modelo o validación.
- Comparar resultados y documentar.

## DEMO PRÁCTICA EN PYTHON

---



Cuaderno de Jupyter



# Gracias



Presentación

**Contacto:** aemerinot@gmail.com



Con el Aval de:



# FUNDAMENTOS DE MACHINE LEARNING



⌚ 19h00 – 21h30

⌚ 18 de febrero  
al 3 de marzo

## APRENDERÁS

Comprender los  
 fundamentos del Machine  
Learning y su lógica principal

Preparar y transformar  
 datos para modelos reales  
en Python

Aplicar modelos de  
 aprendizaje supervisado y  
no supervisado

Analizar patrones y  
 resultados con enfoque  
práctico

**INSCRÍBETE**

📞 +593 98 334 9634

✉️ capacitacion@see-ec.org

