# Descenso de Gradiente en Ciencias Biomédicas

## Regresión lineal múltiple y regresión logística

PhD. Pablo Eduardo Caicedo Rodríguez

2026-01-14

# Objetivos de la sesión

- Entender el **principio del descenso de gradiente (GD)** como método de optimización.
- Aplicar GD a **regresión lineal múltiple** (objetivo continuo) y **regresión logística** (clasificación 0/1).
- Analizar decisiones de ingeniería: **tasa de aprendizaje**, **escalado/normalización**, **batch vs. mini-batch vs. SGD**.
- Interpretar resultados en **contextos biomédicos** (diagnóstico, pronóstico y evaluación de riesgo).

# Agenda

1. Fundamentos de GD y funciones de costo
2. Caso 1: **Regresión lineal múltiple** con GD
3. Caso 2: **Regresión logística** con GD
4. Buenas prácticas, diagnósticos y discusión aplicada

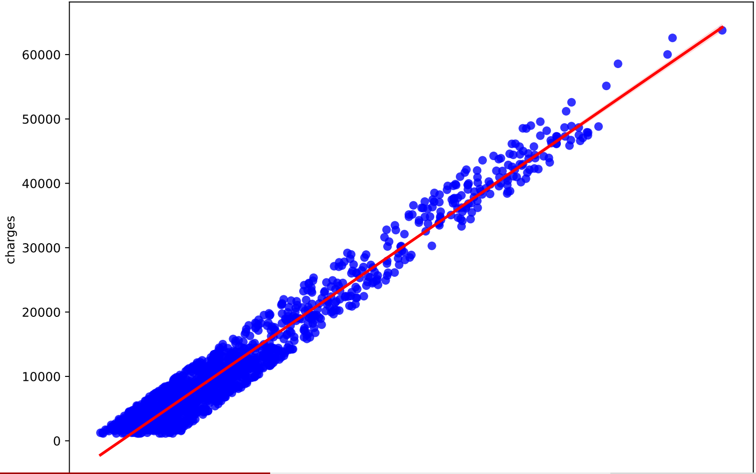
Sección 1

# 1. Fundamentos de descenso de gradiente

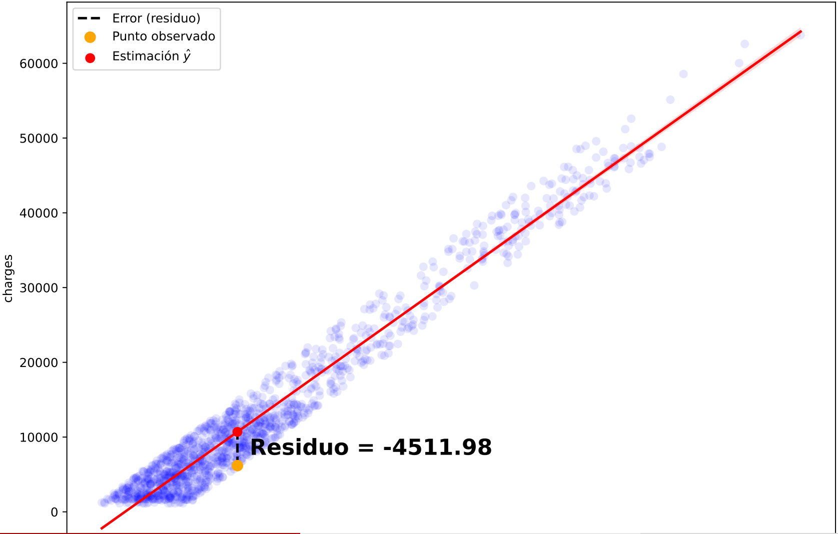
# Motivación biomédica

- Ajustar parámetros de un **modelo fisiológico** o un **predictor clínico** con datos reales.
- Ejemplos típicos:
  - Estimar **gasto energético** a partir de IMC, edad y FC.
  - Estimar **edad vascular** con variables de laboratorio.
  - **Clasificar** presencia/ausencia de una condición (0/1) con biomarcadores.

# Regresión Lineal



# Regresión Lineal



# Función de costo (error)

**Regresión (continuo):**

$$J(\mathbf{w}) \; = \; \frac{1}{2m}\sum_{i=1}^{m}\left(y_j - \hat{y}_j\right)^2$$

donde $\hat{y}_j = \mathbf{w}^\top \mathbf{x}_j$.

**Clasificación binaria: Entropía cruzada** (log-loss):

$$J(\mathbf{w}) \; = \; -\frac{1}{m}\sum_{i=1}^{m}\left[\, y_i \log \hat{p}_i + (1 - y_i)\right.$$

donde $\hat{p}_i = \sigma(\mathbf{w}^\top \mathbf{x}_i) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}_i}}$.

# Función de costo (error)

**! ... Para el caso de ejemplo (un modelo tipo lineal)**

$$\mathbf{w_j}^\top = \left[w_{j,1}, w_{j,0}\right]$$

$$salary_i = x_i$$

$$\hat{y}_j = w_{j,1} * x_j + w_{j,0}$$

$$J(\mathbf{w}) = \frac{1}{2m}\sum_{i=1}^{m}\left(y_j - w_{j,1} * x_i - w_{j,0}\right)^2$$

# Idea central del GD

- Partimos de $\mathbf{w}_{(0)}$ (p. ej., aleatorio).
- Iteramos:

$$\mathbf{w}_{(t+1)} = \mathbf{w}_{(t)} - \alpha \nabla_{\mathbf{w}} J(\mathbf{w}_{(t)})$$

- $\alpha$ = **tasa de aprendizaje**: grande $\to$ rápido pero inestable; pequeña $\to$ estable pero lento.
- Convergencia: buscamos $\nabla J \approx \mathbf{0}$.

**Visualización conceptual:** "valle" del error y trayectoria en zig-zag hacia el mínimo.

# Algoritmo de Descenso de Gradiente

**Para la estimación de $w_{j,0}$**

$$w_{j+1,0} = w_{j,0} - \alpha \frac{\partial J}{\partial w_{j,0}}$$

$$\frac{\partial J}{\partial w_{j,0}} = \frac{\partial}{\partial w_{j,0}} \left( \frac{1}{2m} \sum_{i=1}^{m} \left(y_i - w_{j,1}x_i - w_{j,0}\right)^2 \right)$$

$$\frac{\partial}{\partial w_{j,0}} \left( \frac{1}{2m} \sum_{i=1}^{m} \left(y_i - w_{j,1}x_i - w_{j,0}\right)^2 \right) = \frac{1}{m} \sum_{i=1}^{m} \left(w_{j,1}x_i + w_{j,0} - y_i\right)$$

**Para la estimación de $w_{j,1}$**

# Variantes: batch, mini-batch, SGD

- **Batch GD:** usa todo el conjunto en cada actualización (costo alto por iteración).
- **Mini-batch GD:** usa lotes pequeños (compromiso eficiencia/ruido).
- **SGD (estocástico):** actualiza con una sola muestra por paso (barato, ruidoso, puede escapar de óptimos pobres).

**Práctica recomendada:** mini-batch (p. ej., 32–256).

# Preprocesamiento y escalado

- **Estandarizar** o **normalizar** las $x_j$ acelera y estabiliza GD.
- Centrar: $x_j \leftarrow (x_j - \mu_j)/\sigma_j$.
- Manejo de outliers y transformaciones (log, Box–Cox) cuando aplique.

# 1.5 EDA

Empezar a usar jupyter.

# Sección 2

## 2. Caso 1 · Regresión lineal múltiple con GD

# Planteamiento

**Objetivo biomédico (ejemplo):** predecir **gasto energético** (kcal) a partir de **edad**, **IMC** y **FC**.

Modelo lineal:

$$\hat{y} = \mathbf{w}^\top \mathbf{x} = w_0 + w_1 x_1 + \cdots + w_p x_p$$

Costo (MSE):

$$J(\mathbf{w}) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

Gradiente:

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)\, x_{ij}$$

Actualización:

$$w_j \leftarrow w_j - \alpha \, \frac{\partial J}{\partial w_j}$$

# Pseudocódigo (mini-batch)

```
in: X (m×p), y (m),  , batch_size, epochs
preprocess: X ← standardize(X)

initialize w ← zeros(p+1)  # incluye sesgo w0 si se usa X con col

for epoch in 1..epochs:
    for B in iterate_minibatches(X, y, batch_size, shuffle=True):
        Xb, yb ← B
        yhat ← Xb · w
        grad ← (1/|B|) · (Xb  · (yhat − yb))
        w ← w −   · grad

return w
```

# Diagnóstico y validación

- Curva $J$ vs. iteraciones (entrenamiento y validación).
- Errores residuales: homocedasticidad, estructura vs. predicción.
- Interpretación clínica de coeficientes $w_j$ y unidades.
- Comparar con **ecuaciones normales** (solución cerrada) y discutir condicionamiento numérico.

Sección 3

# 3. Caso 2 · Regresión logística con GD (clasificación 0/1)

# Planteamiento

**Objetivo biomédico (ejemplo):** clasificar **riesgo de enfermedad** (0/1) con panel de biomarcadores.

Modelo:

$$\hat{p} = \sigma(\mathbf{w}^\top \mathbf{x}), \quad \sigma(z) = \frac{1}{1 + e^{-z}}$$

Costo (entropía cruzada):

$$J(\mathbf{w}) = -\frac{1}{m} \sum_{i=1}^{m} \left[ y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i) \right]$$

Gradiente:

$$\frac{\partial J}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} (\hat{p}_i - y_i)\, x_{ij}$$

Actualización:

$\partial J$

# Pseudocódigo (mini-batch)

```
in: X (m×p), y {0,1}^m,  , batch_size, epochs
preprocess: X ← standardize(X)

initialize w ← zeros(p+1)

for epoch in 1..epochs:
    for B in iterate_minibatches(X, y, batch_size, shuffle=True):
        Xb, yb ← B
        z ← Xb · w
        p ← sigmoid(z)
        grad ← (1/|B|) · (Xb  · (p − yb))
        w ← w −   · grad

return w
```

# Métricas y curvas

- **AUC-ROC**, **AUPRC**, **sensibilidad**, **especificidad**, **F1**.
- Elección del **umbral** $\tau$ por criterio clínico (p. ej., maximizar sensibilidad bajo límite de FPs).
- Calibración: curvas de confiabilidad.

# Visualización de la frontera de decisión

- Con dos características $(x_1, x_2)$, la frontera es una **línea** (hiperplano en general).
- Durante GD, la frontera rota/traslada hasta estabilizarse.
- Añadir **términos polinomiales** o **bases** para fronteras no lineales; GD sigue aplicando.

Sección 4

# 4. Buenas prácticas y discusión aplicada

# Hiperparámetros y trucos prácticos

- **Tasa de aprendizaje ($\alpha$)**: búsqueda en rejilla o **programación de tasa** (decay).
- **Inicialización**: pequeña aleatoria (cero puede estancar con ciertas variantes).
- **Barajado** por época y **mini-batches** estratificados si la clase es rara.
- **Regularización** (L2/L1) para estabilidad e interpretabilidad:

$$J_\lambda = J + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \quad \text{(Ridge)}$$

- **Detección de fuga de datos** y validación por **sujeto** en estudios clínicos.

# Checklist de la sesión (rápido)

- ☐ Estandarizaste variables de entrada.
- ☐ Definiste costo adecuado (MSE vs. CE).
- ☐ Elegiste mini-batch y $\alpha$ razonables.
- ☐ Verificaste convergencia con curva de $J$.
- ☐ Evaluaste con métricas adecuadas al objetivo clínico.
- ☐ Documentaste supuestos y limitaciones.

# Objetivos de aprendizaje

- **Comprender** los fundamentos de la **Regresión Lineal**, **Regresión Logística** y **Perceptrón Multicapa (MLP)**.
- **Aplicar** estos modelos al contexto de **salud fetal** con datos de **cardiotocografía (CTG)**.
- **Evaluar** el desempeño con métricas adecuadas (MSE, AUC/Log-Loss, matriz de confusión, F1).

1. Regresión Lineal
2. Regresión Logística
3. Perceptrón Multicapa
4. Cierre y discusión

**Dataset**: `fetal_health.csv` (UCI CTG). **Contexto clínico**: interpretación de CTG (*normal*, *sospechoso*, *patológico*).

# Contexto clínico (CTG)

**! Defición**

Prueba médica que monitoriza simultáneamente la frecuencia cardíaca del feto y la actividad contráctil del útero. Se realiza generalmente durante el tercer trimestre del embarazo y el parto, colocando dos transductores externos (uno para la frecuencia cardíaca fetal y otro para las contracciones) sobre el abdomen de la madre

# Contexto clínico (CTG)

| Característica (Variable en CSV) | Cálculo o Descripción |
|---|---|
| Parámetros Basales | |
| baseline value | Es la frecuencia cardíaca fetal (FCF) media aproximada en un segmento de 10 minutos, excluyendo aceleraciones, deceleraciones y períodos de variabilidad marcada (>25 lpm). Se redondea a incrementos de 5 latidos por minuto (lpm).[4, 5, 6, 7, 8] El rango normal se considera entre 110 y 160 lpm.[9, 10] |
| fetal_movement | Número de movimientos fetales detectados por segundo.[1, 11, 12] |
| uterine_contractions | Número de contracciones uterinas por segundo. Se considera normal tener 5 o menos contracciones en 10 minutos.[1, 4, 11, 12] |

# Contexto clínico (CTG)

- CTG registra **FCF** y **contracciones uterinas**.
- Clasificación clínica (FIGO): **normal / sospechoso / patológico**.
- Variabilidad, aceleraciones y desaceleraciones son claves.

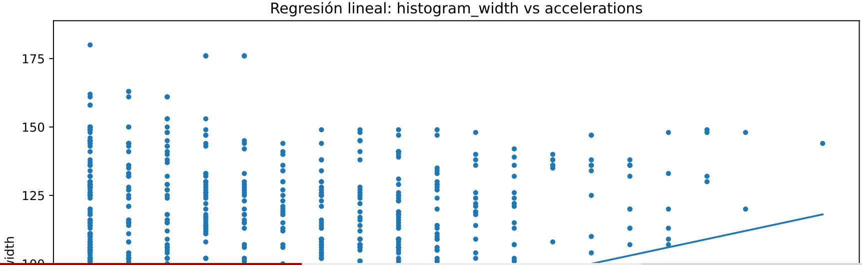
# Regresión Lineal

## Idea clave

Aproxima una relación **lineal** $\hat{y} = \beta_0 + \sum_j \beta_j x_j$ minimizando **MSE**.

## Ejemplo didáctico (CTG)

Usamos una **variable continua** de CTG como respuesta (p. ej., *histogram_width*) para ilustrar ajuste y residuales.



# Regresión Logística

**Idea clave**

Modela $P(Y = 1 \mid \mathbf{x}) = \sigma(\beta_0 + \mathbf{x}^\top \beta)$ con **sigmoide** $\sigma(z) = 1/(1 + e^{-z})$.

# 1. Definición de Regresión Logística

La **Regresión Logística** es un algoritmo de aprendizaje automático supervisado utilizado fundamentalmente para problemas de **clasificación binaria**.

A pesar de su nombre, su objetivo no es predecir un valor continuo, sino modelar la **probabilidad** ($P$) de que una observación pertenezca a una clase específica (usualmente denotada como $Y = 1$).

El modelo toma variables de entrada (features) $x_1, \ldots, x_n$ y estima $P(Y = 1|\mathbf{x})$.

# 2. El Mecanismo Central del Modelo

El modelo logístico opera en dos pasos cruciales:

## 2.1. El Componente Lineal (Logit)

Primero, el modelo calcula una suma ponderada de las entradas, exactamente igual que en una regresión lineal. A este resultado ($z$) se le conoce como **logit** o, más formalmente, **log-odds**.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

- $\beta_0$ es el intercepto (sesgo).
- $\beta_{1\ldots n}$ son los coeficientes (pesos) que el modelo aprende.
- El rango de salida de $z$ es el de todos los números reales: $(-\infty, +\infty)$.

## 2.2. La Función Sigmoide (Logística)

Dado que una probabilidad debe estar en el rango $[0, 1]$, $z$ no puede ser el

# 3. La Relación Clave: Probabilidad y Log-Odds

El concepto central que conecta el modelo lineal con la probabilidad es el **log-odds**. Esta transformación es necesaria para mapear un espacio acotado $[0,1]$ a un espacio no acotado $[-\infty, +\infty]$.

## 3.1. De Probabilidad a Log-Odds

La transformación se realiza en dos pasos:

1. **Probabilidad ($P$)**: La probabilidad del evento.
   - Rango: $[0,1]$
2. **Odds (Momios)**: La razón entre la probabilidad de que ocurra $(P)$ y la de que no ocurra $(1-P)$.
   $$Odds = \frac{P}{1-P}$$
   - Rango: $[0, +\infty]$
3. **Log-Odds (Logit)**: El logaritmo natural de los *odds*.

# 4. Interpretación de Coeficientes

Debido a esta relación, los coeficientes ($\beta_i$) del modelo tienen una interpretación específica:

- **Coeficiente** $\beta_i$: Un incremento de una unidad en la variable $x_i$ (manteniendo las demás constantes) genera un cambio de $\beta_i$ en el **log-odds** de la predicción.
- **Odds Ratio (OR)**: Para una interpretación más intuitiva, se utiliza $e^{\beta_i}$. Un incremento de una unidad en $x_i$ **multiplica** los *odds* por un factor de $e^{\beta_i}$.