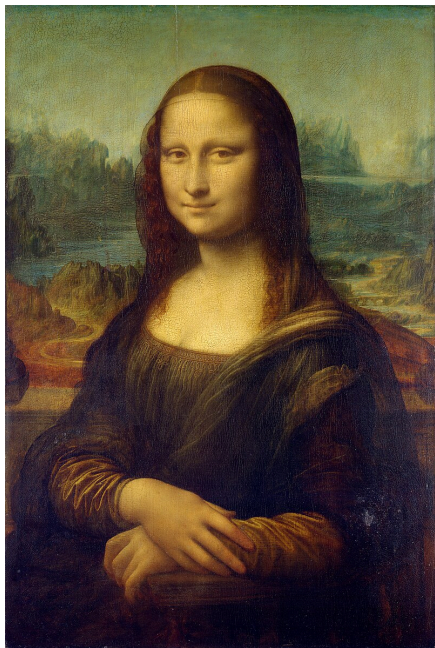


# A Caccia di Parametri

Caso Studio: Posizionare la Gioconda al Louvre



## Il problema

Il quadro più famoso del mondo deve essere appeso alla parete. A che altezza ( $h$ ) posizioniamo il bordo inferiore?

**Vincolo:** il centro del dipinto deve trovarsi **56 cm sopra** l'altezza del visitatore medio.

## I dati disponibili

- Database: **10 milioni** di ingressi anonimi all'anno
- Variabile misurata: altezza dei visitatori
- Popolazione eterogenea: turisti da tutto il mondo, scolaresche, famiglie...

## La domanda

Come estraiamo **un singolo numero** da milioni di misurazioni per prendere una decisione concreta?

# A Caccia di Parametri — La Soluzione

## 1. I Dati (Input)

**Database:** 10 milioni di ingressi anonimi all'anno.

**Variabile:** altezza dei visitatori.

In AI questo è il nostro **Dataset di Addestramento**.

## 2. Il Peso ( $w$ )

Calcoliamo la media delle altezze:

$$m = \frac{1}{N} \sum_{i=1}^N x_i = 175 \text{ cm}$$

Se i dati cambiano, il valore  $m$  cambia.

In AI questo è il **Peso** ( $w$ ): il valore numerico imparato dai dati.

## 3. Il Modello

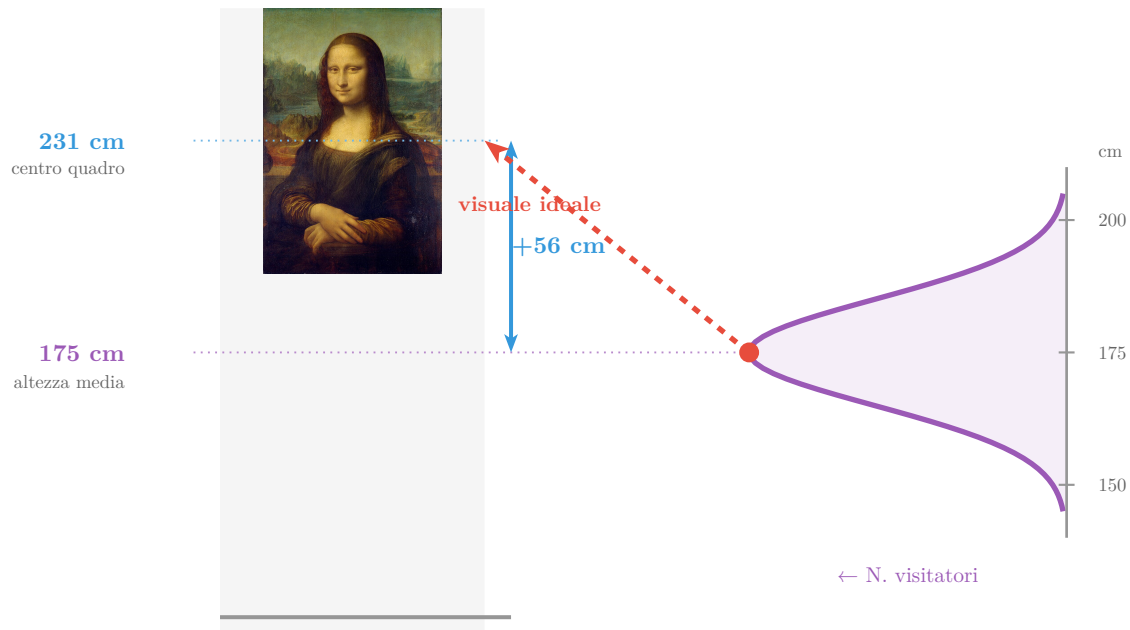
La formula:  $h = m + 56$

È la struttura fissa. Non cambia al variare dei visitatori, ma ospita il peso per produrre il risultato.

In AI questa è la **Funzione di Predizione**: l'architettura del modello.

Il **Modello** è la domanda che poniamo ai dati ( $h = \dots + 56$ ).  
Il **Peso** è la risposta che i dati ci danno ( $m = 175$ ) per agire nel mondo reale.

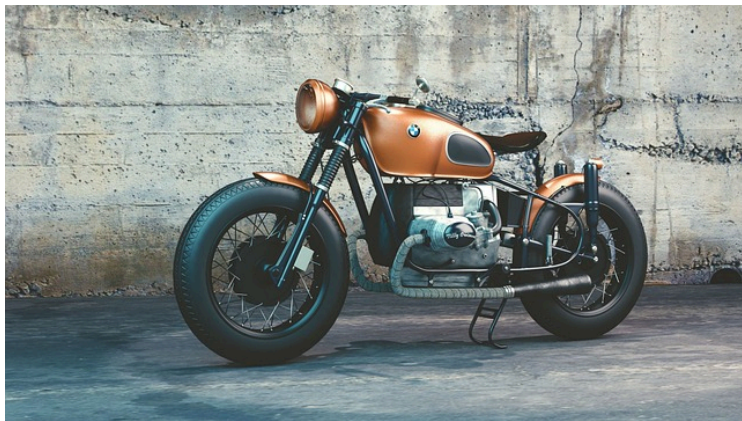
# Perché la Media?



La media ( $m = 175$  cm) è il valore con la **massima frequenza** di visitatori: il miglior compromesso per posizionare il quadro.

# A Caccia di Parametri

Caso Studio: La Moto di Marco



## La promessa

I genitori di Marco gli hanno promesso:

*«Se prendi **9** in fisica, ti compriamo la moto.»*

## Il dilemma di Marco

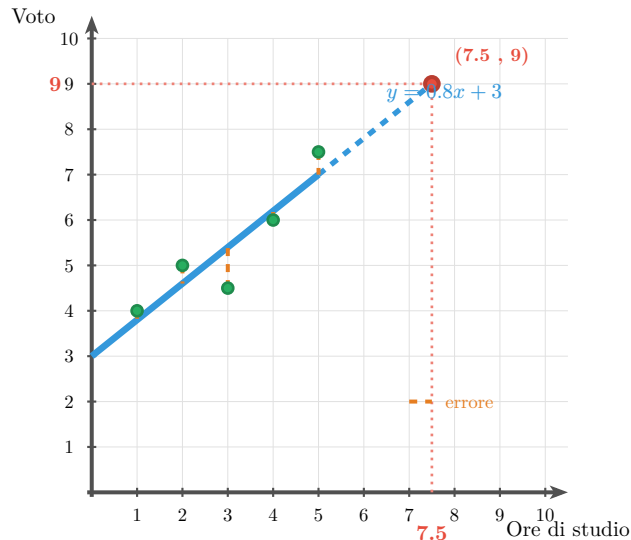
- Non vuole studiare **un minuto di meno** → rischierebbe la moto
- Non vuole studiare **un minuto di più** → efficienza!

Ha i voti dei compiti precedenti. Sa quante ore ha studiato per ciascuno.

## La domanda

Quante ore deve studiare **esattamente** per prendere 9?

# A Caccia di Parametri — La Moto di Marco



## 1. I Dati di Marco

	C1	C2	C3	C4	C5
Ore ( $x$ )	1	2	3	4	5
Voto ( $y$ )	4.0	5.0	4.5	6.0	7.5

## 2. Il Modello (2 parametri)

$$y = mx + q$$

Interpolazione lineare sui dati:

$$m = 0.8 \quad (\text{pendenza})$$

$$q = 3.0 \quad (\text{intercetta})$$

## 3. La Previsione

$$\text{Per ottenere } y = 9: \quad 9 = 0.8x + 3 \rightarrow x = \frac{9-3}{0.8} = 7.5 \text{ ore}$$

In AI: l'Inferenza

# Come Trovare $m$ e $q$

Metodo analitico (troppo difficile per Marco!)

**Passo 1 — Loss** Errore  $e_i = y_i - (mx_i + q)$ . Minimizziamo:

$$L(m, q) = \sum_{i=1}^N (y_i - mx_i - q)^2$$

**Passo 2 — Derivate parziali = 0**

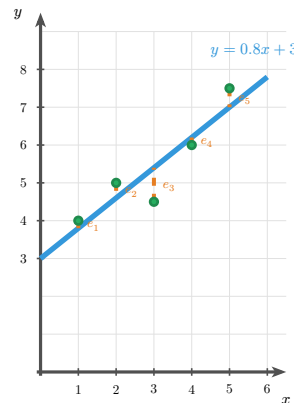
$$\frac{\partial L}{\partial q} = -2 \sum_{i=1}^N (y_i - mx_i - q) = 0$$

$$\frac{\partial L}{\partial m} = -2 \sum_{i=1}^N x_i (y_i - mx_i - q) = 0$$

**Passo 3 — Soluzione**

$$m = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2}$$

$$q = \bar{y} - m\bar{x}$$



Con i dati di Marco  $N = 5$ ,  $\bar{x} = 3$ ,  $\bar{y} = 5.4$   $\rightarrow$   $m = 0.8$   $q = \bar{y} - m\bar{x} = 5.4 - 2.4 = 3.0$

# Come Trovare $m$ e $q$

## Metodo 2 — Massima Verosimiglianza

### Ipotesi — Ogni voto è una gaussiana

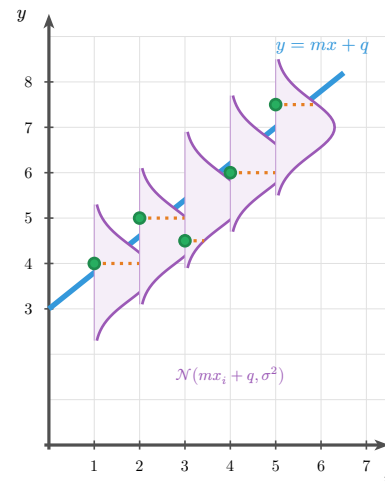
Marco non prende sempre lo stesso voto studiando le stesse ore: c'è variabilità naturale. Modelliamo ogni voto come una variabile casuale gaussiana centrata sulla retta:

$$y_i \sim \mathcal{N}(mx_i + q, \sigma^2)$$

- Il **valore atteso** del voto è sulla retta:  $mx_i + q$
- La **varianza**  $\sigma^2$  misura quanto i voti fluttuano attorno alla retta
- Ogni voto è la **realizzazione** di una gaussiana centrata sulla retta

### La domanda chiave

Quali valori di  $m$  e  $q$  rendono **più probabile** aver osservato esattamente questi voti?



# Massima Verosimiglianza — La Matematica

## Passo 1 — Verosimiglianza (Likelihood)

I voti sono indipendenti, la probabilità congiunta è il **prodotto**:

$$\mathcal{L}(m, q) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - mx_i - q)^2}{2\sigma^2}\right)$$

## Passo 2 — Passaggio ai logaritmi

Il logaritmo trasforma il prodotto in somma:

$$\ln \mathcal{L} = \underbrace{-\frac{N}{2} \ln(2\pi\sigma^2)}_{\text{costante}} - \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - mx_i - q)^2$$

Massimizzare  $\ln \mathcal{L}$  rispetto a  $m$  e  $q$  equivale a **minimizzare** il secondo termine.

## Passo 3 — Conclusione

Massimizzare  $\ln \mathcal{L}$

$\leftrightarrow$

Minimizzare

$$\sum_{i=1}^N (y_i - mx_i - q)^2 = L(m, q)$$

$\rightarrow$  **Stessa Loss!**

L'ipotesi di **errori gaussiani** giustifica il metodo dei **minimi quadrati**. Le due strade portano allo stesso risultato.



# Come Trovare $m$ e $q$

## Metodo 3 — L'Analogia delle Molle

### L'esperimento mentale

Collegiamo ogni punto alla retta con una **molla verticale**. La retta è libera di ruotare ( $m$ ) e traslare ( $q$ ). Si ferma nella posizione di **equilibrio**: energia totale minima.

### La fisica: legge di Hooke

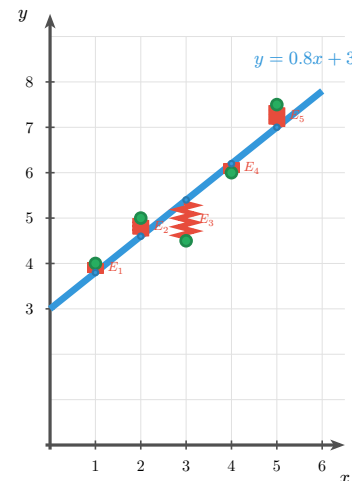
$$E_i = \frac{1}{2}k(\Delta y_i)^2 = \frac{1}{2}k(y_i - mx_i - q)^2$$

Il **quadrato** non è una scelta arbitraria: viene dalla fisica!

### Energia totale = Loss

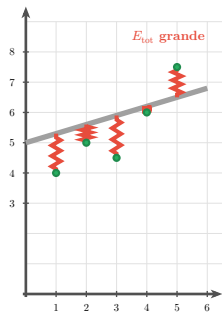
$$E_{\text{tot}} = \frac{1}{2}k \sum_{i=1}^N (y_i - mx_i - q)^2 \propto L(m, q)$$

Minima energia  $\leftrightarrow$  Minimi quadrati

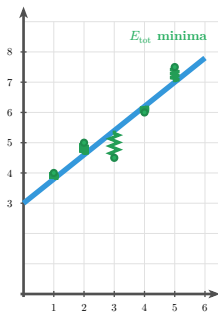


# L'Analogia delle Molle — Equilibrio

Retta sbagliata: molle tese



Retta ottimale: equilibrio



## Condizioni di equilibrio

La retta si ferma quando la **forza netta** è nulla in entrambe le direzioni:

**Traslazione** (regola  $q$ ): la somma delle forze si annulla

$$\sum_{i=1}^N (y_i - mx_i - q) = 0$$

**Rotazione** (regola  $m$ ): la somma dei momenti si annulla

$$\sum_{i=1}^N x_i (y_i - mx_i - q) = 0$$

## Soluzione

$$m = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^N (x_i - \bar{x})^2}$$

$$q = \bar{y} - m\bar{x}$$

Stesse formule dei metodi 1 e 2!

# Come Trovare $m$ e $q$

## Metodo 4 — Discesa del Gradiente

### L'idea — Scendere nella valle

Invece di risolvere equazioni, partiamo da valori qualsiasi di  $m$  e  $q$  e li miglioriamo **passo dopo passo**, scendendo lungo la superficie della Loss:

$$L(m, q) = \sum_{i=1}^N (y_i - mx_i - q)^2$$

### Il gradiente indica la direzione di salita

$$\frac{\partial L}{\partial m} = -2 \sum_{i=1}^N x_i (y_i - mx_i - q) \quad \frac{\partial L}{\partial q} = -2 \sum_{i=1}^N (y_i - mx_i - q)$$

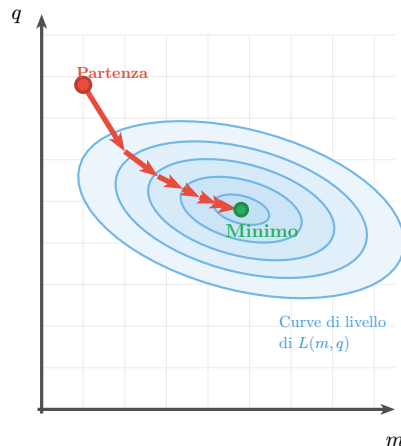
Il gradiente punta in salita → noi camminiamo in **direzione opposta**.

### Regola di aggiornamento

$$m \leftarrow m - \alpha \frac{\partial L}{\partial m}$$

$$q \leftarrow q - \alpha \frac{\partial L}{\partial q}$$

$\alpha$  = learning rate (dimensione del passo) — Ripetere fino a convergenza



# A Caccia di Parametri

Che Cifra È?



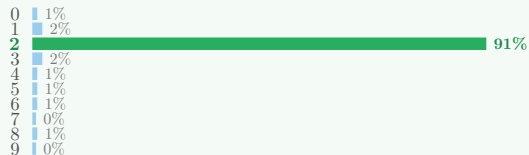
Immagine  $28 \times 28$  pixel  
(784 numeri da 0 a 255)

## Il problema

Ogni immagine è una griglia di pixel, ciascuno con un valore di grigio. Servono **parametri** e un **modello matematico** capace di predire di che cifra si tratta.

## L'obiettivo — Assegnare probabilità

Il modello deve associare all'immagine una **probabilità** per ciascuna delle 10 cifre:



# Riconoscere Cifre con Vettori e Cosine Similarity

## 1. L'immagine è un vettore

Ogni immagine  $28 \times 28$  viene **appiattita**: 784 pixel  $\rightarrow \mathbf{x} \in \mathbb{R}^{784}$

## 2. I parametri: le 10 cifre medie

Per ogni cifra  $d \in \{0, \dots, 9\}$ , calcoliamo la **media** di tutte le immagini di training di quella cifra:

$$\mu_d = \frac{1}{N_d} \sum_{i: y_i = d} \mathbf{x}_i$$



Le 10 cifre medie (parametri del modello):  $10 \times 784 = 7840$  parametri

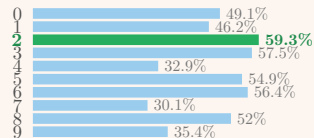
## 3. Predizione:

calcoliamo  $\mathbf{x} \sim \mu_d$  per ogni cifra e scegliamo la più alta:  $\hat{y} = \operatorname{argmax}_d(\mathbf{x} \sim \mu_d)$

## Esempio — Immagine di test



Confrontiamo con le 10 medie:



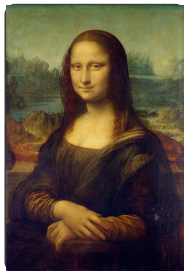
**Predizione: 2** (cos similarity più alta: 59.3%)

## Risultato

**Accuratezza: 82.2%**

8216 su 10000 immagini di test classificate correttamente, usando solo la **media** e la **cosine similarity** — nessuna rete neurale!

# Riepilogo — A Caccia di Parametri



## Dati

10 milioni di altezze dei visitatori



## Dati

5 coppie (ore studio, voto)



## Dati

60 000 immagini  $28 \times 28$

## Parametri

1

La media  $m$  delle altezze

## Parametri

2

Pendenza  $m$  e intercetta  $q$

## Parametri

7 840

10 cifre medie  $\times$  784 pixel

## Modello

$$h = m + 56$$

## Modello

$$y = mx + q$$

## Modello

$$\hat{y} = \operatorname{argmax}_d (x \sim \mu_d)$$