

# Lezione 2: Machine Learning in Elettronica

## Caratterizzazione V-I e Regressione Lineare

Laboratorio di Sistemi e Automazione

### 1 Obiettivo del Laboratorio

In elettronica, spesso dobbiamo determinare il valore di un componente sconosciuto analizzando il suo comportamento. Oggi useremo la **\*\*Regressione Lineare\*\*** per calcolare la resistenza  $R$  di un componente partendo da una serie di misure sperimentali di Corrente ( $I$ ) e Tensione ( $V$ ), affette da errore.

### 2 Il Modello Fisico: La Legge di Ohm

Sappiamo dalla teoria che per un conduttore ohmico vale la relazione lineare:

$$V = R \cdot I$$

Confrontandola con l'equazione generica della retta ( $y = mx + q$ ):

- **Input ( $x$ ):** Corrente  $I$  (in milliAmpere, mA).
- **Target ( $y$ ):** Tensione  $V$  (in Volt, V).
- **Pendenza ( $m$ ):** Resistenza  $R$  (in  $k\Omega$ ).
- **Intercetta ( $q$ ):** Errore di offset (o tensione di bias).

### 3 Fase 1: I Dati "Rumorosi" (Input)

Abbiamo effettuato 5 misure in laboratorio usando un multimetro non calibrato perfettamente. I dati non sono allineati.

Corrente $I$ (mA)	Tensione $V$ (Volt) - Misurata
1.0	1.1
2.0	1.9
3.0	3.2
4.0	3.8
5.0	5.1

## 4 Fase 2: Il Dialogo con Python

Non calcoliamo  $R = V/I$  punto per punto (otterremmo 5 valori diversi!). Usiamo l'AI per trovare la "miglior retta possibile" che media gli errori.

```
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3
4 # 1. PREPARAZIONE DATI
5 # X (Input): Corrente in mA. NOTA: Servono le doppie parentesi [[ ]]
6 X_corrente = np.array([[1.0], [2.0], [3.0], [4.0], [5.0]])
7
8 # y (Target): Tensione in Volt misurata (con errore)
9 y_tensione = np.array([1.1, 1.9, 3.2, 3.8, 5.1])
10
11 # 2. ADDESTRAMENTO (FITTING)
12 # Chiediamo alla macchina di trovare m (Resistenza) e q (Offset)
13 modello = LinearRegression()
14 modello.fit(X_corrente, y_tensione)
15
16 print("Analisi completata.")
```

Listing 1: Caricamento Dati e Training

## 5 Fase 3: Interpretazione dei Risultati

Abbiamo ottenuto due numeri fondamentali dal nostro modello:

- **Coefficiente angolare ( $m$ ):** 0.98 (Questa è la Resistenza in  $k\Omega$ ).
- **Intercetta ( $q$ ):** 0.05 (Questo è l'errore di offset del multimetro in  $V$ ).

Ma questi numeri sono affidabili?

## 5.1 Che cos'è visivamente la Regressione?

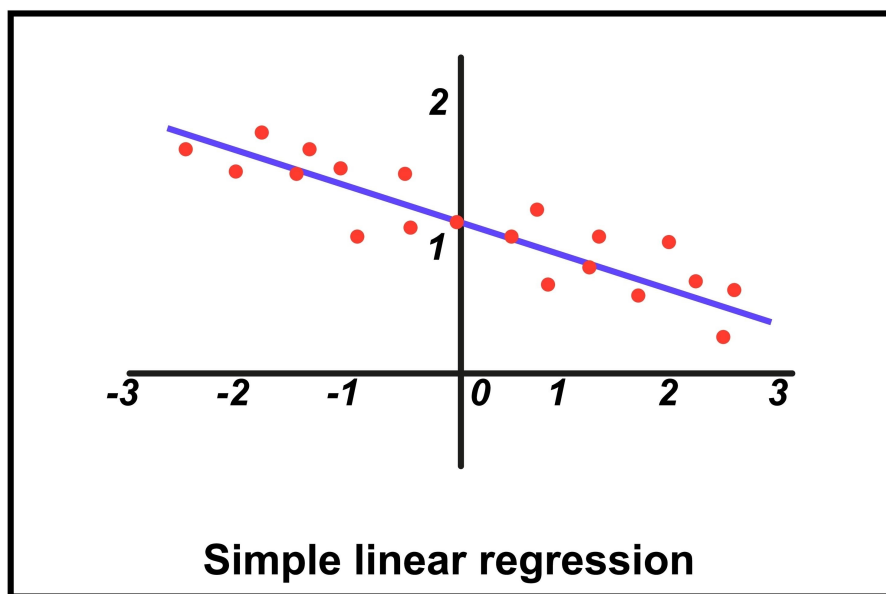


Figura 1: Retta di regressione interpretazione geometrica

Se potessimo vedere i nostri dati su un grafico, vedremmo i punti delle misure che non sono perfettamente allineati (a causa del rumore elettrico).

La **\*\*Retta di Regressione\*\*** che abbiamo appena calcolato è quella linea immaginaria che passa "nel mezzo" della nuvola di punti, minimizzando la distanza media da ciascuno di essi. È il miglior compromesso possibile tra tutti gli errori di misura.

## 6 Prossimo Step: La Visualizzazione

I numeri da soli non bastano a un tecnico. Per confermare che il sensore o il resistore funzioni bene, dobbiamo vedere il grafico **Caratteristica V-I**.

Nella prossima lezione impareremo a usare la libreria grafica `matplotlib` per:

1. Disegnare i punti sperimentali (Scatter Plot).
2. Sovrapporre la retta che abbiamo appena calcolato.
3. Vedere a occhio nudo quanto è preciso il componente.

→ **CONTINUA CON LA PARTE GRAFICA**

Il calcolo è fatto. Ora passiamo al disegno tecnico.  
Clicca qui sotto per andare alla lezione sulla visualizzazione:

**Lezione Python\_3: Rea-  
lizzazione dei Grafici**