

# Lezione 3: Visualizzazione Dati

## Creazione della Caratteristica V-I e Salvataggio File

Laboratorio di Sistemi e Automazione

## 1 Introduzione

Nella lezione precedente abbiamo calcolato la resistenza ( $R$ ) e l'offset. Ora dobbiamo creare il grafico tecnico per la relazione.

### Avviso per Google Colab

Se hai chiuso il notebook o ricaricato la pagina dopo la Lezione 2, la memoria si è svuotata!

Prima di fare i grafici, devi rieseguire l'importazione dei dati e delle librerie.

## 2 Fase 0: Setup dell'Ambiente (Importante)

Per evitare errori come `NameError: name 'plt' is not defined` o dati mancanti, eseguiamo questa cella per preparare tutto il necessario.

```
1 # 1. Importiamo TUTTE le librerie necessarie
2 import matplotlib.pyplot as plt # Per i grafici
3 import numpy as np # Per i calcoli numerici
4 from sklearn.linear_model import LinearRegression # Per il modello
5
6 # 2. Riconfiguriamo i dati della Lezione 2 (Misure Lab)
7 X_corrente = np.array([[1.0], [2.0], [3.0], [4.0], [5.0]])
8 y_tensione = np.array([1.1, 1.9, 3.2, 3.8, 5.1])
9
10 # 3. Ricalcoliamo il modello (Fit)
11 modello = LinearRegression()
12 modello.fit(X_corrente, y_tensione)
13
14 print("Setup completato: Librerie caricate e Modello pronto.")
```

Listing 1: Setup Iniziale (Eseguire per primo!)

## 3 Fase 1: Lo Scatter Plot (I Punti)

Il primo passo è visualizzare le nostre misure "grezze". Usiamo `plt.scatter` che disegna i punti senza collegarli.

```
1 plt.figure(figsize=(8, 5))
2
```

```

3 # Disegniamo i punti blu (Misure)
4 plt.scatter(X_corrente, y_tensione, color='blue', label='Misure Lab')
5
6 # Aggiungiamo la legenda per capire cos'è il punto blu
7 plt.legend()
8 plt.show()

```

Listing 2: Visualizzare le misure sperimentali

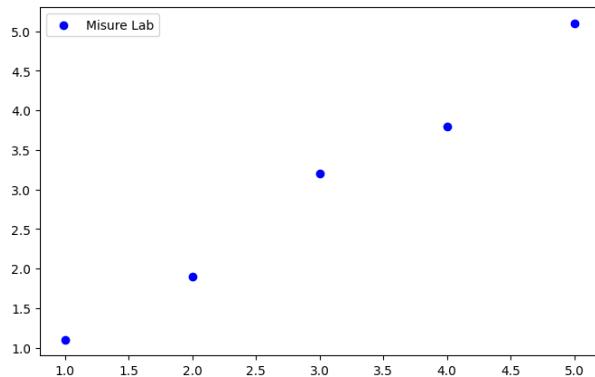


Figura 1: Rappresentazione dei punti della nostra tabella

## 4 Fase 2: Sovrapporre la Regressione

Ora disegniamo la linea rossa che rappresenta il comportamento "ideale" calcolato dall'AI.

```

1 # 1. Calcoliamo i valori teorici ideali (sulla retta)
2 y_predetti = modello.predict(X_corrente)
3
4 plt.figure(figsize=(8, 5))
5
6 # 2. Rimettiamo i punti blu (Scatter)
7 plt.scatter(X_corrente, y_tensione, color='blue', label='Misure Lab')
8
9 # 3. Aggiungiamo la LINEA rossa (Plot)
10 # 'r-' significa: colore rosso (r), linea continua (-)
11 plt.plot(X_corrente, y_predetti, 'r-', linewidth=2, label='Modello Calcolato')
12
13 plt.legend()
14 plt.show()

```

Listing 3: Aggiungere la retta di regressione

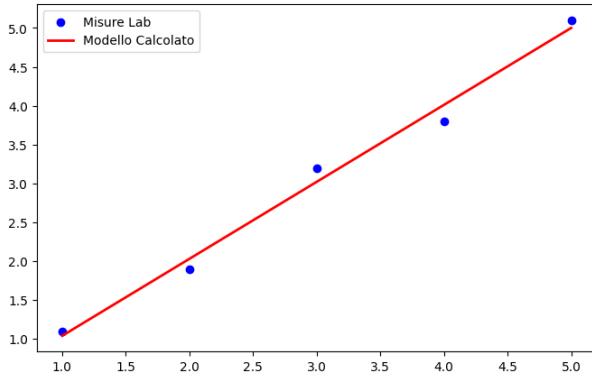


Figura 2: Calcolo retta di regressione

## 5 Fase 3: Il Grafico Tecnico Completo

Un grafico da presentare in una relazione tecnica deve essere auto-esplicativo. Aggiungiamo Titolo, Etichette Assi (con unità di misura!) e Griglia.

```

1 plt.figure(figsize=(10, 6)) # Creiamo un foglio più grande
2
3 # --- DISEGNO ---
4 plt.scatter(X_corrente, y_tensione, color='blue', s=80, label='Misure Sperimentali')
5 plt.plot(X_corrente, y_predetti, color='red', linewidth=3, label='Caratteristica V-I ('
6           'Regessione)')
7
8 # --- ESTETICA (Labelling) ---
9 plt.title("Caratteristica V-I Componente Ohmico", fontsize=14)
10 plt.xlabel("Corrente I [mA]", fontsize=12)
11 plt.ylabel("Tensione V [Volt]", fontsize=12)
12
13 # --- DETTAGLI ---
14 plt.grid(True, linestyle='--', alpha=0.7) # Griglia tratteggiata
15 plt.legend(fontsize=12)                  # Legenda
16
17 # --- OUTPUT ---
18 plt.show()

```

Listing 4: Grafico Finale Professionale

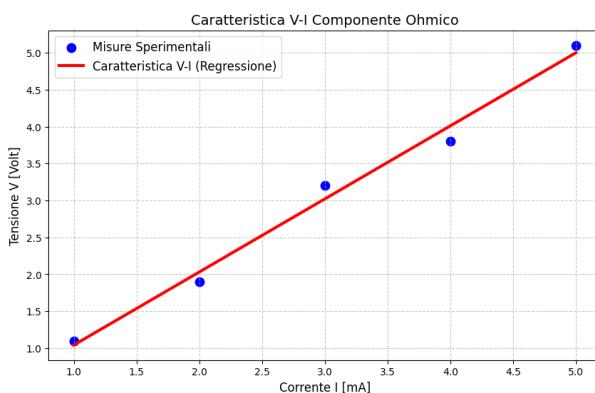


Figura 3: Enter Caption

## 6 Fase 4: Salvare il Grafico su File

Per inserire il grafico nella tua relazione, devi salvarlo come immagine.

Nota su Colab

Dopo aver eseguito questo codice, apri la cartella **Files** (icona della cartella a sinistra dello schermo) per trovare e scaricare l'immagine `grafico_VI.png`.

```
1 # rieseguiamo il codice del grafico (senza plt.show)
2 plt.figure(figsize=(10, 6))
3 plt.scatter(X_corrente, y_tensione, color='blue', label='Misure')
4 plt.plot(X_corrente, y_predetti, color='red', label='Modello')
5 plt.title("Caratteristica V-I")
6 plt.xlabel("Corrente [mA]")
7 plt.ylabel("Tensione [V]")
8 plt.grid(True)
9 plt.legend()
10
11 # COMANDO DI SALVATAGGIO
12 # dpi=300 garantisce alta risoluzione per la stampa
13 plt.savefig("grafico_VI.png", dpi=300)
14
15 print("Immagine salvata! Controlla la cartella Files.")
```

Listing 5: Salvataggio in PNG

→ RISORSA ONLINE

Vuoi vedere il codice in azione senza digitare tutto?  
Clicca qui sotto per aprire il Notebook Colab già pronto:

[ ESECUZIONE PROGRAMMA IN COLAB ]

## 7 E ora? Apriamo la Scatola Nera

Abbiamo visto che la macchina traccia una linea perfetta. Ma come fa a sceglierla? Perché non ha tracciato una linea un po' più in alto o un po' più in basso?

Nella prossima lezione scopriremo il "cervello" matematico dell'algoritmo: l'errore quadratico medio e il concetto di apprendimento.

→ CONTINUA IL PERCORSO

Capiamo come "ragiona" l'algoritmo.  
Vai alla Lezione 4: I Fondamenti del Machine Learning

**Lezione Python\_4: La Teoria dell'Errore**