PROBLEMA 2 — ANALISI MATRICI DI CONFUSIONE (DOCUMENTAZIONE)

Versione: 1.0  
Data: 20-08-2025 13:53:58  
Autori: Yurii Riina, Nicolò Gioacchini, Thomas Marinucci

# INDICE

1) Obiettivo del problema

2) Panoramica della soluzione (cosa fa, per chi, perché)

3) Requisiti (MATLAB/toolbox) e file di progetto

4) Come avviare l’app (“Quick Start”)

5) Architettura e flusso dei dati

6) Interfaccia utente (tab e funzionalità)

7) Algoritmo di calcolo metriche

8) Gestione normalizzazione e confronti

9) Input supportati e come testarli

10) Output generati (UI, CSV, sessioni, log)

11) Gestione sessioni, cronologia e log

12) Gestione errori e casi limite

13) Struttura delle cartelle (consigliata/attesa)

14) Mappatura ai requisiti della consegna

15) Estensioni future suggerite

16) Crediti e note finali

# 1) Obiettivo del problema

Implementare in MATLAB un sistema di analisi di matrici di confusione per valutare le prestazioni di classificatori.  
- Accettare input da file (.mat o .csv)  
- Visualizzare la matrice con heatmap interattiva  
- Calcolare metriche (accuratezza, precision, recall, F1, balanced accuracy)  
- Consentire confronti fra matrici  
- Salvare risultati e log

# 2) Panoramica della soluzione (cosa fa, per chi, perché)

La soluzione fornisce un’APP MATLAB con interfaccia moderna che permette di:  
• Caricare matrici NxN da file .mat/.csv  
• Visualizzare matrici in forma numerica e grafica  
• Normalizzare per riga o globalmente  
• Calcolare metriche globali e per classe  
• Confrontare diverse matrici  
• Salvare risultati, log e immagini  
  
Target: studenti, docenti e valutatori che vogliono verificare la bontà di un classificatore.  
Perché così: la GUI consente analisi rapida, trasparente e riproducibile.

# 3) Requisiti (MATLAB/toolbox) e file di progetto

Requisiti:  
• MATLAB R2021a o superiore  
• Nessuna toolbox obbligatoria oltre a funzioni base (imagesc, uitable, etc.)  
  
File principali:  
- UI: confusionMatrixUI.m  
- Logic: metriche (accuracy, precision, recall, f1), normalizzazione  
- Utils: writeFullLog.m, logMessage.m, toggleFAQEntry.m  
- Integrazione: createApp.m, launchModule.m

# 4) Come avviare l’app (“Quick Start”)

Quick Start:  
1. Aprire MATLAB nella radice del progetto  
2. Eseguire createApp  
3. Nella tab Progetti scegliere “📊 Problema 2 – Analisi matrici di confusione”

# 5) Architettura e flusso dei dati

Architettura:  
- loadMatrixFromFile → importa da .mat/.csv  
- plotConfusionMatrix → genera heatmap su Axes  
- computeMetrics → calcola accuracy, precision, recall, F1  
- compareMatrices → confronto lato Tab 3  
- Sessione: gestione storico matrici + log esteso

# 6) Interfaccia utente (tab e funzionalità)

Interfaccia (5 Tab):  
TAB 1 – Info Modulo: obiettivi e descrizione  
TAB 2 – Matrice di confusione: caricamento, heatmap, normalizzazione, opzioni  
TAB 3 – Metriche & Confronto: calcolo metriche e confronto matrici diverse  
TAB 4 – Sessione: storico matrici analizzate, salvataggi CSV, log esteso  
TAB 5 – Guida: FAQ espandibili/collassabili

# 7) Algoritmo di calcolo metriche

Algoritmo metriche:  
- Accuratezza globale: sum(diag(C)) / sum(C(:))  
- Per-classe: acc\_i = C(i,i) / sum(C(i,:))  
- Precision(i) = C(i,i) / sum(C(:,i))  
- Recall(i) = C(i,i) / sum(C(i,:))  
- F1(i) = 2 \* (Prec\*Rec) / (Prec+Rec)  
- Balanced Accuracy = mean(recall\_i)

# 8) Gestione normalizzazione e confronti

Normalizzazione:  
- Per riga: ogni riga divisa per somma della riga (per confronti su dataset sbilanciati)  
- Globale: diviso per somma totale elementi  
  
Confronti:  
- L’utente può selezionare due matrici dallo storico (Tab 3) e confrontarle graficamente

# 9) Input supportati e come testarli

Input supportati:  
- File .mat con variabile C (NxN) e opzionalmente labels  
- File .csv con header opzionale  
- Demo interne (caricate dalla GUI)

# 10) Output generati (UI, CSV, sessioni, log)

Output generati:  
- UI: heatmap, tabelle metriche  
- CSV: export dei risultati metriche  
- Figure: salvataggio heatmap  
- Log: dettagliato con timestamp

# 11) Gestione sessioni, cronologia e log

Gestione sessioni/log:  
- writeFullLog: messaggi dettagliati con timestamp  
- logMessage: log sintetico  
- Sessione salva cronologia matrici, metriche calcolate, log

# 12) Gestione errori e casi limite

Errori e casi limite:  
- File non leggibile → alert  
- Matrice non quadrata → errore bloccante  
- Righe a zero → NaN nelle metriche  
- Confronto con meno di due matrici → alert

# 13) Struttura delle cartelle (consigliata/attesa)

Struttura cartelle consigliata:  
Radice progetto/  
 ├─ Main/  
 │ ├─ createApp.m  
 │ └─ ModulesUI/Problem\_2\_UI/confusionMatrixUI.m  
 ├─ Modules/Problem\_2/ (logic, utils)  
 ├─ Data/Problem\_2/sessions, exports

# 14) Mappatura ai requisiti della consegna

Mappatura requisiti:  
- Input: .mat/.csv → OK  
- Normalizzazione per riga → OK  
- Accuratezza per classe e globale → OK  
- Confronto matrici → OK  
- Salvataggi/log → OK  
- FAQ con domande frequenti → OK

# 15) Estensioni future suggerite

Estensioni future:  
- Aggiungere più metriche (AUC, MCC, Cohen’s Kappa)  
- Supporto a classificatori multi-label  
- Integrazione con Problem 1 per validazione gesti  
- Esportazione diretta in LaTeX/Excel

# 16) Crediti e note finali

Crediti:  
Sviluppo: Yurii Riina, Nicolò Gioacchini, Thomas Marinucci  
Corso: Approcci e Sistemi di Interfacciamento per i Videogame e la Realtà Virtuale  
Anno: 2024/2025  
Dipendenze: MATLAB base  
  
Supporto: usare la tab “Guida” o consultare Documentation/