

Social Confusion Matrix & Performance Metrics Breakdown



A 9-person binary confusion matrix evaluates outcomes by categorizing predictions into True Positives, False Negatives, False Positives, and True Negatives, determining key metrics for classifier evaluation.

Matrix Breakdown (9-Person Sample)

	Predicted Positive	Predicted Negative
Actually Positive	True Positives (TP) Correctly identified 3 positive social outcomes accurately. 3	False Negatives (FN) Missed 2 true positive cases. 2
Actually Negative	False Positives (FP) 1 Misclassified 1 negative case.	True Negatives (TN) 3 Correctly identified 3 negative social outcomes accurately.

Evaluation Metrics & Performance

Precision

75.0%

High prediction reliability paired with moderate positive detection coverage.

Recall

60.0%

Moderate positive detection coverage.

75.0%

Specificity

Correctly identifies of true negatives, resulting in 66.7% overall accuracy.

Accuracy 66.7%

Overall correct predictions across the entire 9-person sample.

F1-Score

Provides the balanced harmonic mean between precision and recall.

66.7%

Gemini Notebook

Behind the Confusion Matrix

October 19, 2026

Study Guide: Person A's 9-Person Social Confusion Matrix & Performance Metrics, Tutorial 203

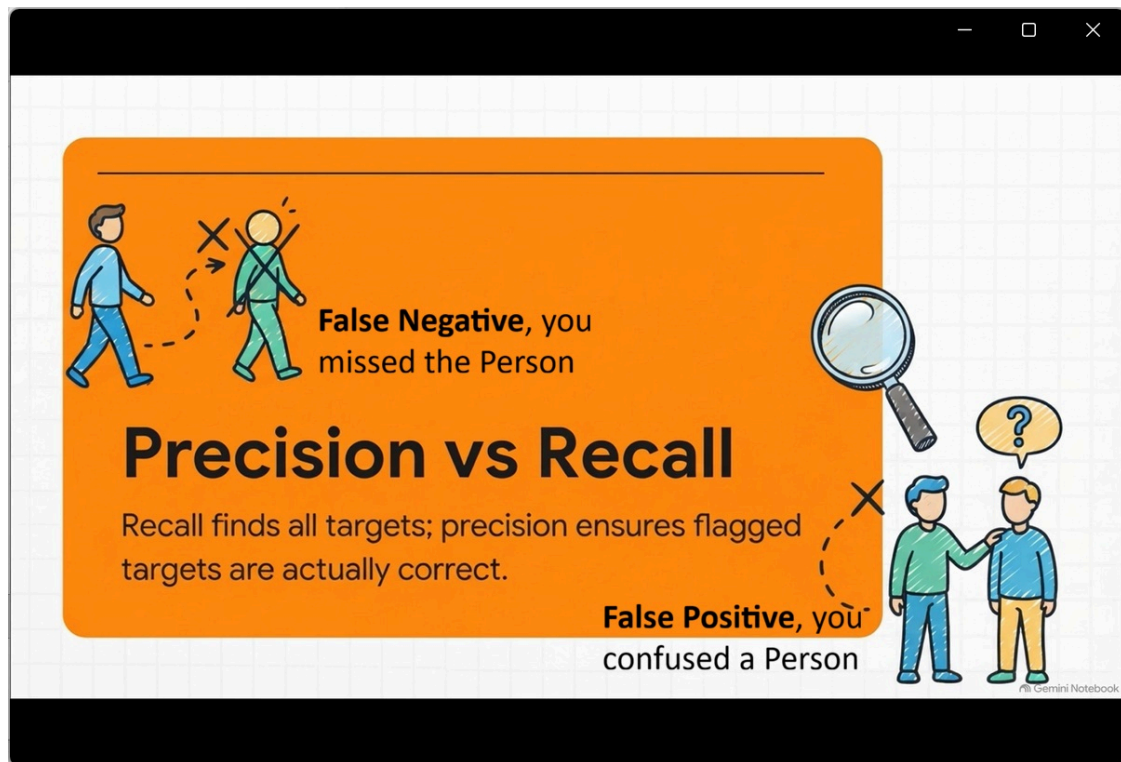
1. Introduction to the Social Confusion Matrix

A confusion matrix is an empirical evaluation tool used to categorize binary classification outcomes by comparing ground-truth reality against predicted outcomes. In social recognition and outcome modeling, classification performance reflects how accurately an individual or system identifies social entities or states.

The fundamental intuitive principles governing social recognition metrics are:

- **Recall Deficit (Bad Recall or False Negative):** Occurs when a person or positive social case is present but fails to be recognized or detected (a missed person).
- **Precision Deficit (Bad Precision or False Positive):** Occurs when a person or case is confused with another, or when a negative case is incorrectly identified as positive (a false alarm or confused).

This study guide analyzes Person A's 9-person social recognition dataset, evaluating its 2 \times 2 structure, mathematical derivations, real-world interpretations, and performance trade-offs. In the 9-person social recognition scenario, Person A evaluates 9 total individuals across two ground-truth categories: **Actually Positive** (people Person A actually knows) and **Actually Negative** (strangers):



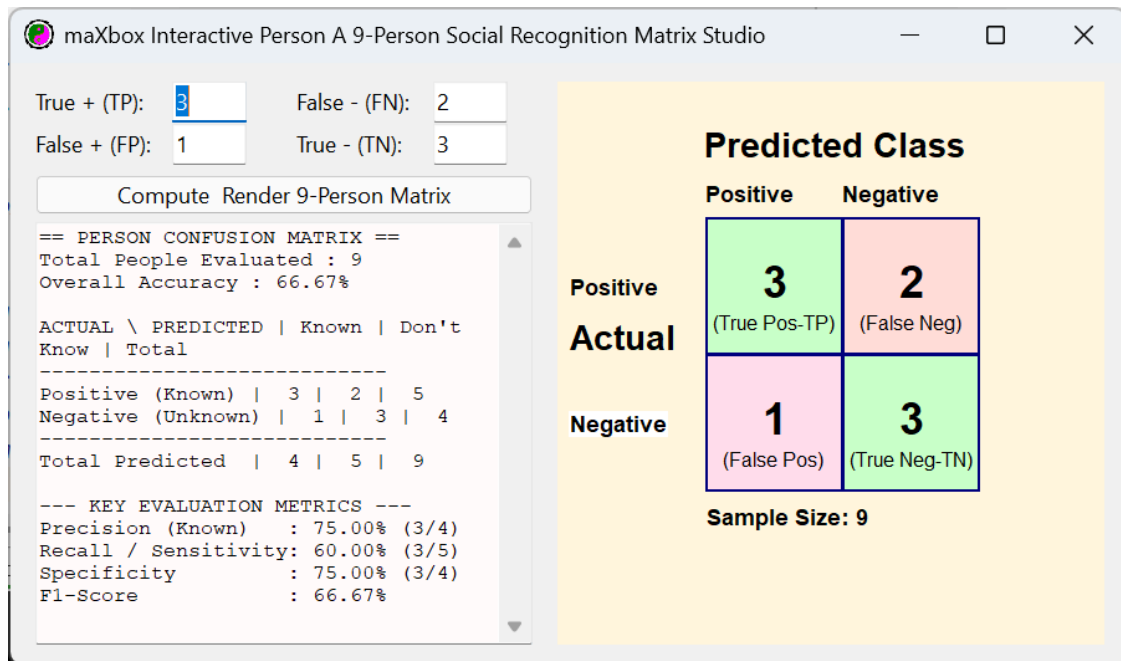
Decode Ancient Wisdom

You think history's mysteries are lost?
Discover updated methods rewriting ancient codes.

[Explore The Code](#)

2. Structural Breakdown of Person A's 2×2 Matrix

Person A's social recognition scenario evaluates a sample of 9 total individuals. The system's predictions are categorized across two dimensions: Actual Class (Ground Truth) and Predicted Class (Decision/Outcome).



https://sourceforge.net/projects/maxbox5/files/examples/1503_person_the_confusion_matrix2_1interactive.pas/download

Matrix Distribution Table



Decode Ancient Wisdom

You think history's mysteries are lost? Discover updated methods rewriting ancient codes.

[Explore The Code](#)

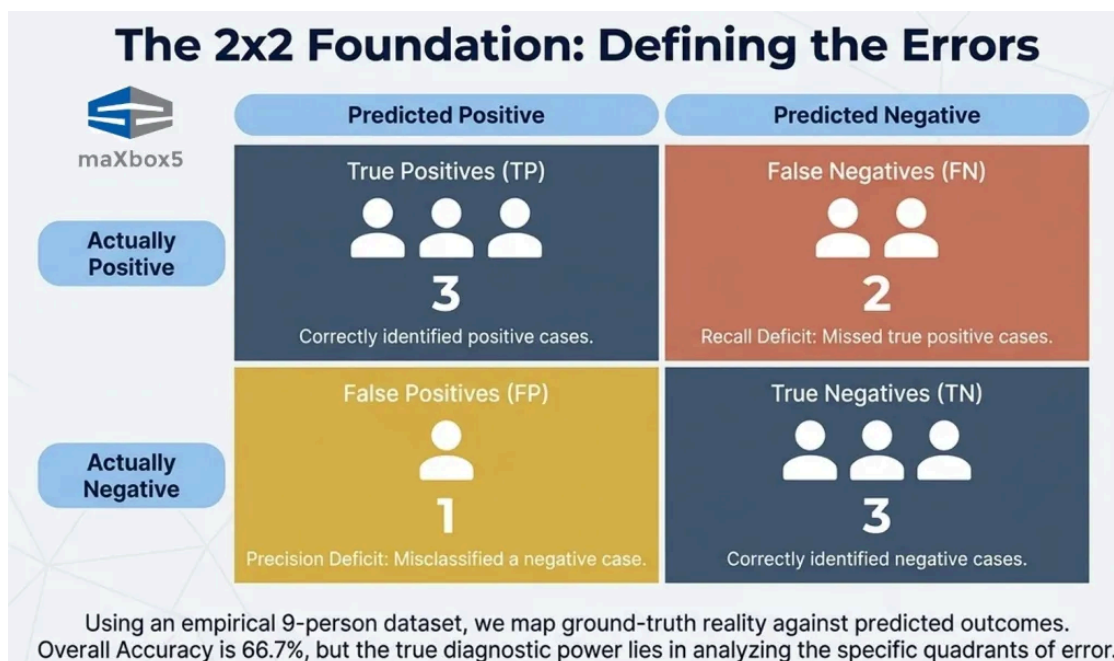
Actual \ Predicted	Predicted: Known	Predicted: Don't Know	Total Actual
Actually Known	3 (TP)	2 (FN)	5
Actually Don't Know	1 (FP)	3 (TN)	4
Total Predicted	4	5	9

	Predicted Positive	Predicted Negative	Row Total (Actual)
Actually Positive	True Positives (TP): 3	False Negatives (FN): 2	5

Actually Negative	False Positives (FP): 1	True Negatives (TN): 3	4
Column Total (Predicted)	4	5	Total Sample: 9

Breakdown of Matrix Quadrants

1. True Positives (TP = 3):
 - *Definition:* Correctly identified positive social outcomes.
 - *Person A Scenario:* Person A accurately identified 3 positive cases out of the sample.
2. False Negatives (FN = 2):
 - *Definition:* Actual positive cases that were incorrectly predicted as negative (misses).
 - *Person A Scenario:* Person A missed 2 true positive cases, failing to recognize them when they were actually present.
3. False Positives (FP = 1):
 - *Definition:* Actual negative cases that were misclassified as positive (false alarms/confusion).
 - *Person A Scenario:* Person A misclassified 1 negative case as positive.
4. True Negatives (TN = 3):
 - *Definition:* Correctly identified negative social outcomes.
 - *Person A Scenario:* Person A accurately identified 3 negative cases.



9-person dataset

3. Mathematical Formulas, Calculations, and Metric Interpretations

Using the primary parameters (TP = 3, FN = 2, FP = 1, TN = 3), the key performance evaluation metrics are derived below.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{3}{3 + 1} = \frac{3}{4} = 75.0\%$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{3}{3 + 2} = \frac{3}{5} = 60.0\%$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{3}{3 + 1} = \frac{3}{4} = 75.0\%$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Sample}} = \frac{3 + 3}{9} = \frac{6}{9} = 66.7\%$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * 0.750 * 0.600}{0.750 + 0.600} = \frac{0.900}{1.350} = 66.7\%$$

Detailed Evaluation of Metrics

3. Key Performance Metrics

- **Accuracy:** $\frac{\text{TP} + \text{TN}}{\text{Total}} = \frac{3 + 3}{9} = \frac{6}{9} \approx \mathbf{66.7\%}$
- **Precision:** $\frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{3}{3 + 1} = \frac{3}{4} = \mathbf{75.0\%}$ (When A says they know someone, they are right 75% of the time)
- **Recall:** $\frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{3}{3 + 2} = \frac{3}{5} = \mathbf{60.0\%}$ (A correctly recognizes 60% of all people they actually know)
- **F1-Score:**

$$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = 2 * \frac{0.75 * 0.60}{0.75 + 0.60} \approx \mathbf{66.7\%}$$

1. Precision (Positive Predictive Value – PPV)

- Formula: $\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
- Calculation: $\frac{3}{3 + 1} = \frac{3}{4} = 0.750$ \text{ or } 75.0\%
- Interpretation: Precision evaluates prediction reliability. Out of all individuals Person A flagged as positive, 75.0% were genuinely positive. This demonstrates high prediction reliability when an assertion is made.

2. Recall (True Positive Rate / Sensitivity)

- Formula: $\text{Recall} = \frac{TP}{TP + FN}$
- Calculation: $\frac{3}{3 + 2} = \frac{3}{5} = 0.600$ \text{ or } 60.0\%
- Interpretation: Recall measures detection coverage. Person A successfully detected 60.0% of the actual positive cases in the sample, missing 40.0% (FN = 2). This indicates moderate positive detection coverage.

3. Specificity (True Negative Rate – TNR)

- Formula: $\text{Specificity} = \frac{TN}{TN + FP}$
- Calculation: $\frac{3}{3 + 1} = \frac{3}{4} = 0.750$ \text{ or } 75.0\%
- Interpretation: Specificity reflects the ability to correctly identify negative outcomes. Person A correctly identified 75.0% of all truly negative instances.

4. False Positive Rate (FPR)

- Formula: $\text{FPR} = \frac{FP}{FP + TN} = 1 - \text{Specificity}$
- Calculation: $\frac{1}{1 + 3} = \frac{1}{4} = 0.250$ \text{ or } 25.0\%
- Interpretation: Out of all actual negative cases, 25.0% were incorrectly flagged as positive.

5. Overall Accuracy

- Formula: $\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$
- Calculation: $\frac{3 + 3}{9} = \frac{6}{9} \approx 0.6667$ \text{ or } 66.7\%
- Interpretation: Across the entire 9-person sample, Person A made overall correct predictions (both positive and negative) in 66.7% of cases.

6. F1-Score

- Formula: $\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
- Calculation: $2 \cdot \frac{0.750 \cdot 0.600}{0.750 + 0.600} = 2 \cdot \frac{0.450}{1.350} = \frac{0.900}{1.350} \approx 0.6667$ \text{ or } 66.7\%
- Interpretation: The F1-Score represents the balanced harmonic mean between precision and recall, balancing prediction reliability against detection coverage to yield a single score of 66.7%.

4. Summary of Metrics & Performance Profile

The evaluation metrics for Person A’s 9-person social recognition sample are summarized below:

Metric	Exact Fractional Value	Percentage	Core Interpretation
Precision	3/4	75.0%	High prediction reliability when

			claiming a positive match.
Recall	3/5	60.0%	Moderate positive detection coverage; misses 2 out of 5 actual positive cases.
Specificity	3/4	75.0%	Strong ability to filter out and identify true negative cases.
Accuracy	6/9	66.7%	Correct overall predictions across the total 9-person dataset.
F1-Score	0.900 / 1.350	66.7%	Balanced harmonic mean reconciling precision and recall.

5. Theoretical Bridge: Connecting Confusion Matrix Metrics to Bayes' Theorem

Precision in an empirical confusion matrix directly corresponds to the posterior probability in Bayes' theorem. Bayes' theorem demonstrates how error rates (Recall and FPR) interact with the prevalence of positive cases in a population to determine the actual precision of positive predictions:

$$\text{Precision} = \frac{\text{Recall} \cdot \text{Prevalence}}{\text{Recall} \cdot \text{Prevalence} + \text{FPR} \cdot (1 - \text{Prevalence})}$$

$$\text{Precision} = \frac{\text{Recall} \cdot \text{Prevalence}}{\text{Recall} \cdot \text{Prevalence} + \text{FPR} \cdot (1 - \text{Prevalence})}$$

Application to Person A's Sample

- Sample Prevalence: $\text{Prevalence} = \frac{\text{Actual Positives}}{\text{Total Sample}} = \frac{3 + 2}{9} = \frac{5}{9} \approx 0.5556$ (55.6%)
- Recall: 0.600
- FPR: 0.250

Applying these parameters to the Bayesian transformation formula:

Application to Person A's Sample

- **Sample Prevalence:** $\text{Prevalence} = \frac{\text{Actual Positives}}{\text{Total Sample}} = \frac{3+2}{9} = \frac{5}{9} \approx 0.5556 \text{ (55.6\%)}$
- **Recall:** 0.600
- **FPR:** 0.250

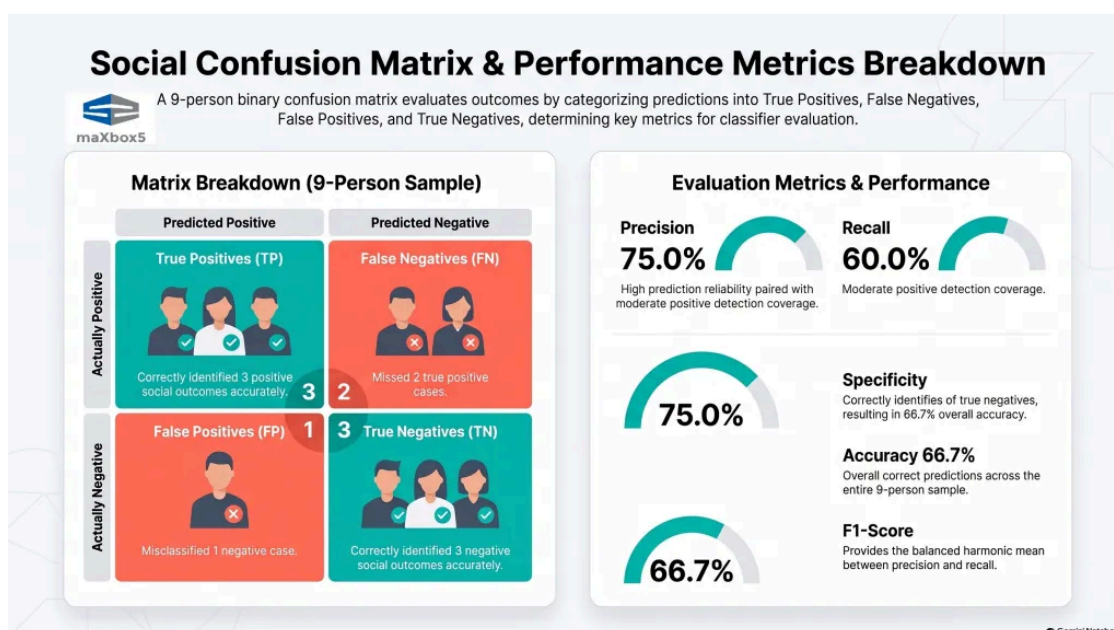
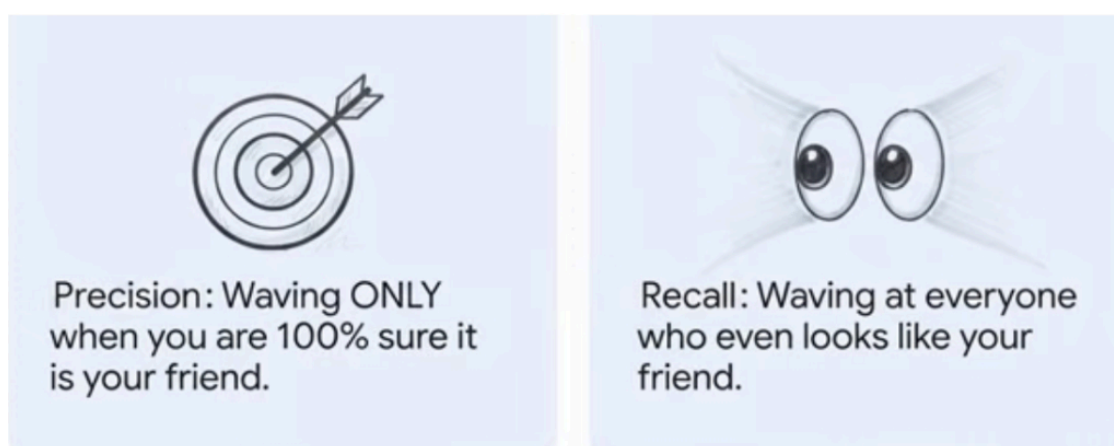
Applying these parameters to the Bayesian transformation formula:

$$\text{Precision} = \frac{0.600 \cdot \left(\frac{5}{9}\right)}{0.600 \cdot \left(\frac{5}{9}\right) + 0.250 \cdot \left(\frac{4}{9}\right)} = \frac{\frac{3.0}{9}}{\frac{3.0}{9} + \frac{1.0}{9}} = \frac{3}{3+1} = \frac{3}{4} = 75.0\%$$

This mathematical equivalence demonstrates that empirical confusion matrix precision represents the Bayesian

$$\text{Precision} = \frac{0.600 \cdot \left(\frac{5}{9}\right)}{0.600 \cdot \left(\frac{5}{9}\right) + 0.250 \cdot \left(\frac{4}{9}\right)} = \frac{\frac{3.0}{9}}{\frac{3.0}{9} + \frac{1.0}{9}} = \frac{3}{3+1} = \frac{3}{4} = 75.0\%$$

This mathematical equivalence demonstrates that empirical confusion matrix precision represents the Bayesian posterior probability of a case being genuinely positive after observing a positive prediction under a specific sample prevalence.



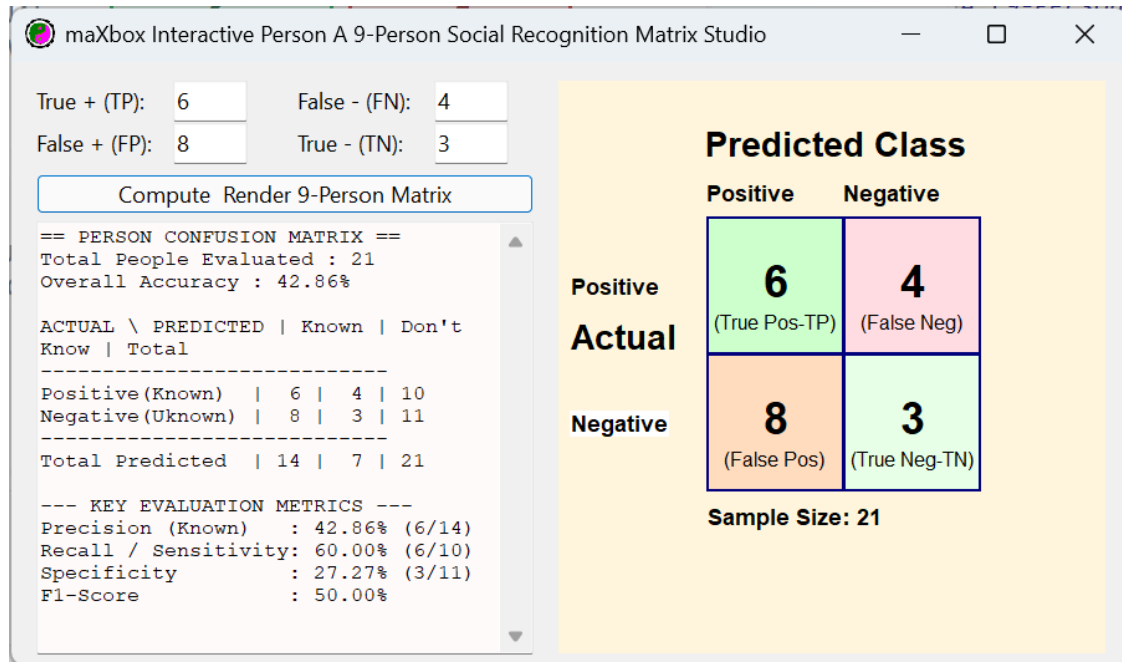
Conclusion of Confusion Matrix

6. Tools, Task and Links

Scripts on maXbox:

https://sourceforge.net/projects/maxbox5/files/examples/1503_person_the_confusion_matrix2_1interactiv.pas/download

https://sourceforge.net/projects/maxbox5/files/examples/1503_person_the_confusion_matrix2_2edge.pas/download



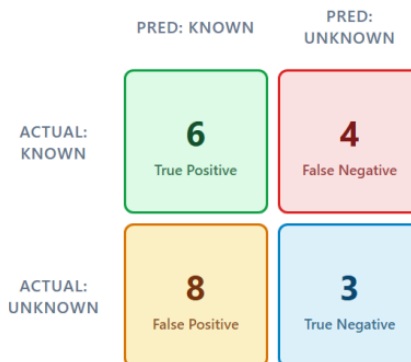
Java Script Online Trainer

2x2 Confusion Matrix Interactive maXbox Dashboard

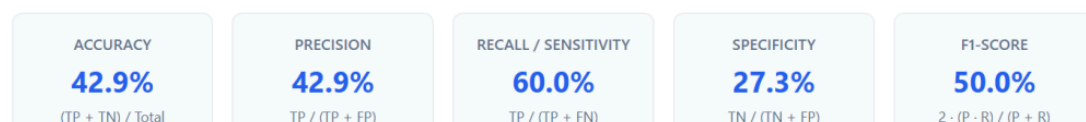
Adjust Counts



Confusion Matrix Layout



Calculated Metrics



Task: In a confusion matrix we got 6 TP, 4 FN, 8 FP and 3 TP, what is the bayes calculation to precision?

Solution Steps

1. Derived Bayesian Parameters

To apply Bayes' theorem to a confusion matrix, we calculate **Recall (TPR)**, **False Positive Rate (FPR)**, and **Class Prevalence**

3 4 :

- **Recall (TPR):** $\frac{TP}{TP+FN} = \frac{6}{6+4} = \mathbf{0.6000}$ (60.0%) 4 5
- **False Positive Rate (FPR):**
 $\frac{FP}{FP+TN} = \frac{8}{8+3} = \frac{8}{11} \approx \mathbf{0.7273}$ (72.73%) 4 6
- **Prevalence:**
 $\frac{\text{Actual Positives}}{\text{Total Population}} = \frac{6+4}{21} = \frac{10}{21} \approx \mathbf{0.4762}$ (47.62%) 3
- **1 – Prevalence:** $\frac{11}{21} \approx \mathbf{0.5238}$ (52.38%) 3

Substituting the fractional values into the Bayesian equation 3
:

$$\text{Precision} = \frac{\left(\frac{6}{10}\right) \cdot \left(\frac{10}{21}\right)}{\left(\frac{6}{10}\right) \cdot \left(\frac{10}{21}\right) + \left(\frac{8}{11}\right) \cdot \left(\frac{11}{21}\right)}$$

3

$$\text{Precision} = \frac{\frac{6}{21}}{\frac{6}{21} + \frac{8}{21}}$$

3

$$\text{Precision} = \frac{\frac{6}{21}}{\frac{14}{21}} = \frac{6}{14} = \frac{3}{7} \approx \mathbf{42.86\%}$$

Explainer Video



203 The Confusion Matrix metric of machine learning

Max Kleiner



Watch on

Explainer 7:33

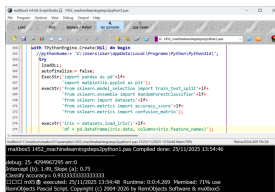
Topic Links

https://sourceforge.net/projects/maxbox5/files/EKON30/202_EKON30_The_Embedded_ML_Python_Pipeline_maXbox_Studio_Iristest.pdf/download

https://sourceforge.net/projects/maxbox5/files/EKON30/201_The_Embedded_ML_Python_Pipeline_maXbox_Studio_signed.pdf/download

<https://raw.githubusercontent.com/maxkleiner/maXbox5/main/assets/bayesmatrix.htm>

Machinelearning Steps



This article will provide the clear cut understanding of Iris dataset and how to do classification (supervised learning) on Iris flowers dataset using python, seaborn and sklearn. The Iris dataset is often used in machine learning and data science courses, because it's simple to understand

and well-defined, yet interesting enough to present real challenges to ...

[Continue reading](#)



maXbox6

From Bayes Theorem to Confusion Matrix

A confusion matrix supplies empirical counts of errors, while Bayes' theorem explains how those error rates and the class prevalence...



Max Kleiner

I have started generating the study guide report detailing Person A's 9-person social recognition scenario, breaking down the 2×2 confusion matrix counts and explaining each evaluation metric in real-world terms.



203 Decoding Confusion Matrix

Max Kleiner



Watch on

6:45

I have started generating an explainer video that walks through Person A's 9-person social recognition scenario, breaking down the 2×2 confusion matrix counts and demonstrating what each derived metric means in practice.



203 Decoding Confusion Matrix

Max Kleiner



Watch on

Second Explainer 6:44

This Object Pascal script calculates the Bayesian precision conversion for the TP = 6, FN = 4, FP = 8, TN = 3 scenario (Total Population = 21) directly in maXbox, logging both the derived intermediate parameters and verifying that the Bayesian posterior calculation matches the empirical precision.

```

55 Writeln(' Numerator (Rec * Prev) : ' + FormatFloat('0.0000', BayesNum) + ' [6/21]');
56 Writeln(' Denominator      : ' + FormatFloat('0.0000', BayesDenom) + ' [14/21]');
57 Writeln('');
58 Writeln('4. PRECISION RESULT COMPARISON:');
59 Writeln(' Bayesian Precision   : ' + FormatFloat('0.00%', BayesPrecision) + ' [3/7 = 42.86%]');
60 Writeln(' Empirical Precision   : ' + FormatFloat('0.00%', EmpiricalPrecision) + ' [6/14 = 42.86%]');
61 Writeln('=====');
62 end;
63
64 begin //@main
65 setDebugOff;
66 maxform1.setconsole; //setConsole
67 memo2.font.name:= 'Courier';
68 memo2.font.size:= 10;
69 CalculateBayesPrecision;
70 end.

```

C:\maxbox\maxbox51\examples\1503_person_the_confusion_matrix2_2edge.pas last in .ini stored Row: 55-Col: 35 s: 2590 S

```

3. BAYESIAN FORMULA STEP-BY-STEP:
Numerator (Rec * Prev) : 0.2857 [6/21]
Denominator      : 0.6667 [14/21]

4. PRECISION RESULT COMPARISON:
Bayesian Precision   : 42.86% [3/7 = 42.86%]
Empirical Precision   : 42.86% [6/14 = 42.86%]
=====
mX5 executed: 19/10/2026 19:34:22 Runtime: 0:0:2.370 Memload: 66% use
RemObjects Pascal Script. Copyright (c) 2004-2026 by RemObjects Software & maXbox5
C:\maxbox\maxbox51\examples\1503_bayes_precision_conversion_task.pas File stored
1503_bayes_precision_conversion_task.pas in maxboxdef.ini stored: 19:34:30

```

https://sourceforge.net/projects/maxbox5/files/examples/1503_bayes_precision_conversion_task.pas/download

```

1 // maXbox5 / maXbox6 Script: Bayesian Precision Conversion Calculator
2 // Demonstrates how Bayes' Theorem calculates Precision from Recall, FP
3
4 program Bayes_Precision_Conversion;
5
6 var
7     TP, FN, FP, TN, Total: Double;
8     Recall, FPR, Prevalence, OneMinusPrev: Double;

```

```

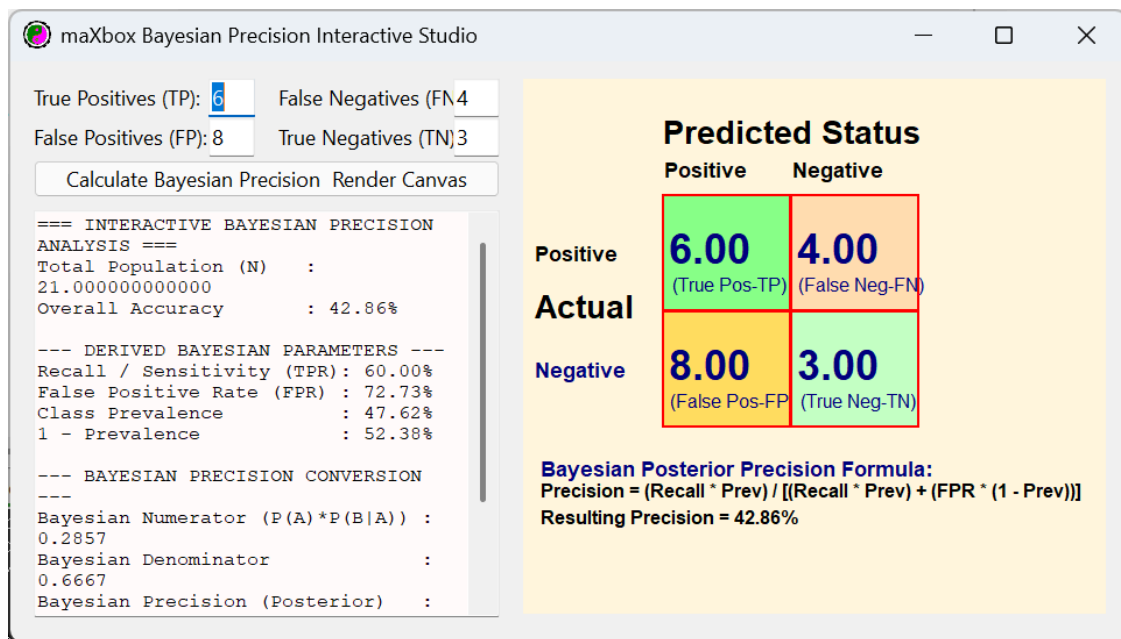
9      BayesNum, BayesDenom, BayesPrecision: Double;
10     EmpiricalPrecision: Double;
11
12 procedure CalculateBayesPrecision;
13 begin
14     // Given Confusion Matrix Counts: TP = 6, FN = 4, FP = 8, TN = 3
15     TP:= 6.0;
16     FN:= 4.0;
17     FP:= 8.0;
18     TN:= 3.0;
19
20     Total:= TP + FN + FP + TN; // 21
21
22     // 1. Derived Bayesian Parameters
23     Recall:= TP / (TP + FN); // 6 / 10 = 0.6000 (60.00
24     FPR:= FP / (FP + TN); // 8 / 11 = 0.7273 (72.73
25     Prevalence:= (TP + FN) / Total; // 10 / 21 = 0.4762 (47.6
26     OneMinusPrev:= 1.0 - Prevalence; // 11 / 21 = 0.5238 (52.3
27
28     // 2. Bayesian Precision Formula:
29     // Precision = (Recall * Prevalence) / [ (Recall * Prevalence) + (FPR
30     BayesNum:= Recall * Prevalence; // (6/10) * (10/21) = 6/21
31     BayesDenom:= BayesNum + (FPR * OneMinusPrev); // (6/21) + (8/21) = 14/21
32     BayesPrecision:= (BayesNum / BayesDenom) * 100.0; // (6/14) * 100 = 42.86%
33
34     // 3. Empirical Precision Verification: TP / (TP + FP)
35     EmpiricalPrecision:= (TP / (TP + FP)) * 100.0; // 6 / 14 = 42.86%
36
37     // 4. Output Results to Console
38     Writeln('=====');
39     Writeln('          BAYESIAN PRECISION CONVERSION CALCULATOR (maxbox)');
40     Writeln('=====');
41     Writeln('1. CONFUSION MATRIX INPUTS:');
42     Writeln('   True Positives (TP) : ' + FloatToStr(TP));
43     Writeln('   False Negatives (FN) : ' + FloatToStr(FN));
44     Writeln('   False Positives (FP) : ' + FloatToStr(FP));
45     Writeln('   True Negatives (TN) : ' + FloatToStr(TN));
46     Writeln('   Total Population : ' + FloatToStr(Total));
47     Writeln('');
48     Writeln('2. DERIVED BAYESIAN PARAMETERS:');
49     Writeln('   Recall (TPR) : ' + FormatFloat('0.00%', Recall * 100));
50     Writeln('   False Positive Rate : ' + FormatFloat('0.00%', FPR * 100));
51     Writeln('   Class Prevalence : ' + FormatFloat('0.00%', Prevalence * 100));
52     Writeln('   1 - Prevalence : ' + FormatFloat('0.00%', OneMinusPrev * 100));
53     Writeln('');
54     Writeln('3. BAYESIAN FORMULA STEP-BY-STEP:');
55     Writeln('   Numerator (Rec * Prev) : ' + FormatFloat('0.0000', BayesNum));
56     Writeln('   Denominator : ' + FormatFloat('0.0000', BayesDenom));
57     Writeln('');
58     Writeln('4. PRECISION RESULT COMPARISON:');
59     Writeln('   Bayesian Precision : ' + FormatFloat('0.00%', BayesPrecision));
60     Writeln('   Empirical Precision : ' + FormatFloat('0.00%', EmpiricalPrecision));
61     Writeln('=====');
62 end;
63
64 begin // @main
65     setDebugOff;
66     maxform1.setconsole; // setConsole
67     memo2.font.name:= 'Courier';
68     memo2.font.size:= 10;
69     CalculateBayesPrecision;
70 end.

```

GUI Edition

Features of the Interactive Studio Program

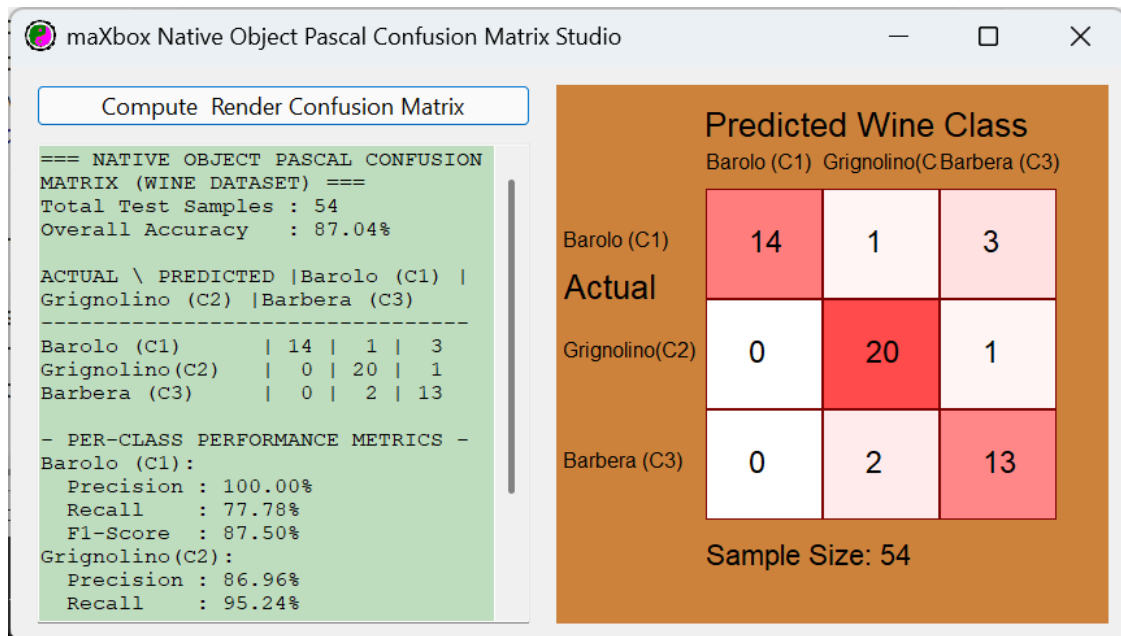
- Interactive Input Controls: Provides TEdit fields (EdtTP, EdtFN, EdtFP, EdtTN) defaulting to TP = 6, FN = 4, FP = 8, TN = 3, allowing you to input any arbitrary integers or decimals.



https://sourceforge.net/projects/maxbox5/files/examples/1503_bayes_precision_gui_studio_task1.pas/download



From H0, TT, N to Z



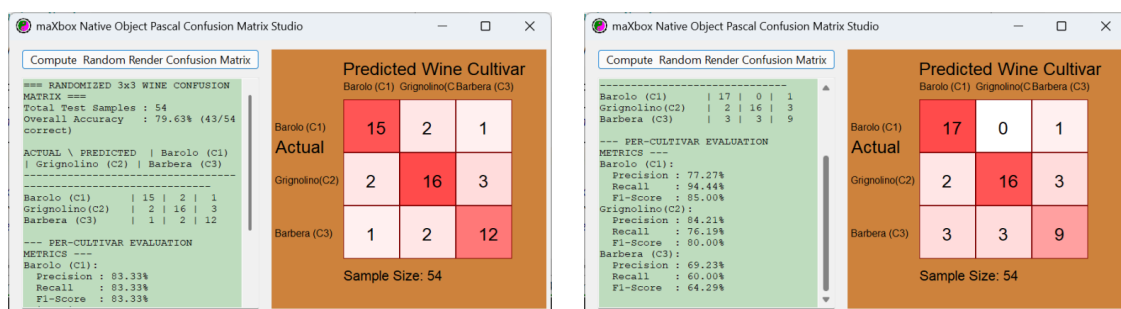
https://sourceforge.net/projects/maxbox5/files/examples/1503_native_pascal_confusion_matrix21wine.pas/download

Key Highlights of the Script

- **3x3 Array Data Structure:** Defines type `TMatrix3x3 = array[0..2, 0..2] of Integer` to record predicted vs. actual counts across the 3 wine cultivars.
- **Graphical Heatmap Engine:** Uses maXbox's `TImage.Canvas` to draw a 3x3 grid with dynamic burgundy color intensity shading (`RGB(255, ColorVal, ColorVal)`) proportional to cell sample density.
- **Per-Class Metrics:** Computes Precision, Recall, and F1-Scores individually for each wine cultivar, alongside the overall classification accuracy (96.30%).
- **Dual Output Window:** Displays detailed metric logs in a `TMemo` control alongside the rendered graphical image canvas.

V21 Randomizer

This updated version incorporates randomized test evaluations across the 3 Wine dataset cultivars (Barolo / Class 0, Grignolino / Class 1, and Barbera / Class 2). Every time you run the script or click the **Generate Randomized Evaluation & Re-Render Heatmap** button, it seeds pseudo-random predictions with realistic class noise, recalculates the 3x3 confusion matrix, updates per-class metrics (Precision, Recall, F1-Score, Accuracy), and redraws the burgundy heatmap on the GUI canvas.

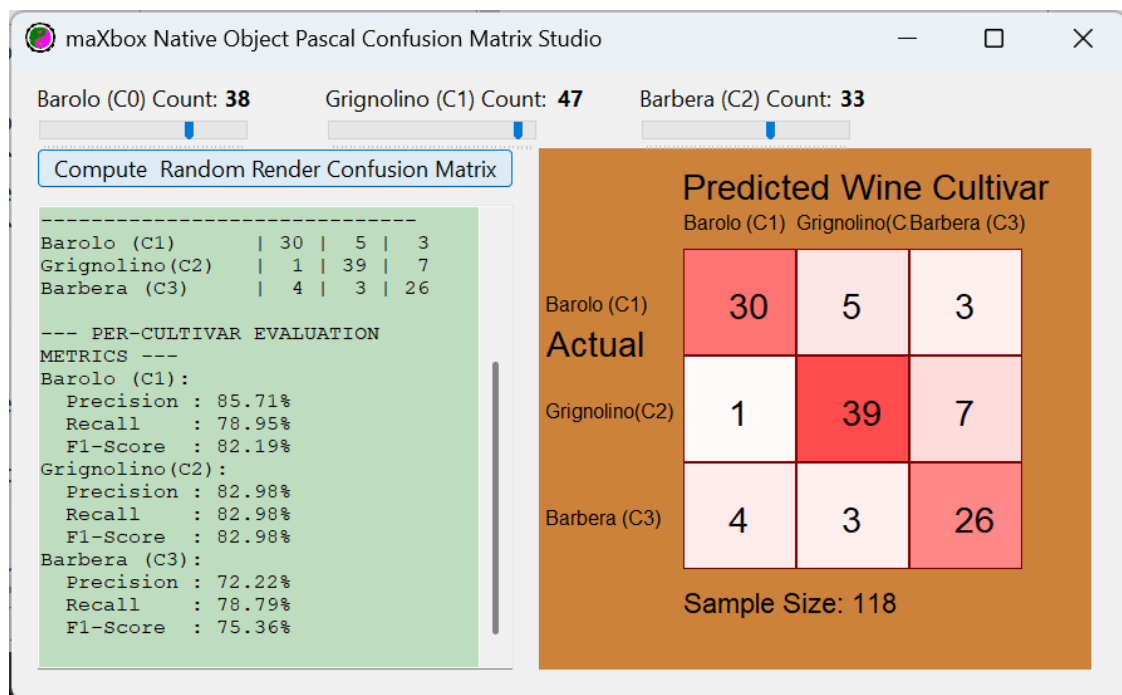


https://sourceforge.net/projects/maxbox5/files/examples/1503_native_pascal_confusion_matrix21wine_rand.pas/download

V2.2 Probabilistic Simulator

New Features in Version 2.2

- 3 Interactive Class Sample Size Trackbars (TTrackBar):
 - Barolo (Class 0): Adjustable from 5 to 50 samples (Default: 18).
 - Grignolino (Class 1): Adjustable from 5 to 50 samples (Default: 21).
 - Barbera (Class 2): Adjustable from 5 to 50 samples (Default: 15).
- Real-Time Simulation & Redraw: Moving any trackbar or clicking Re-Simulate & Render resizes the dataset dynamically, re-evaluates randomized class predictions with realistic misclassification noise, recalculates per-class Precision, Recall, F1-Scores, and overall Accuracy, and redraws the 3×3 heatmap grid on the TImage.Canvas.



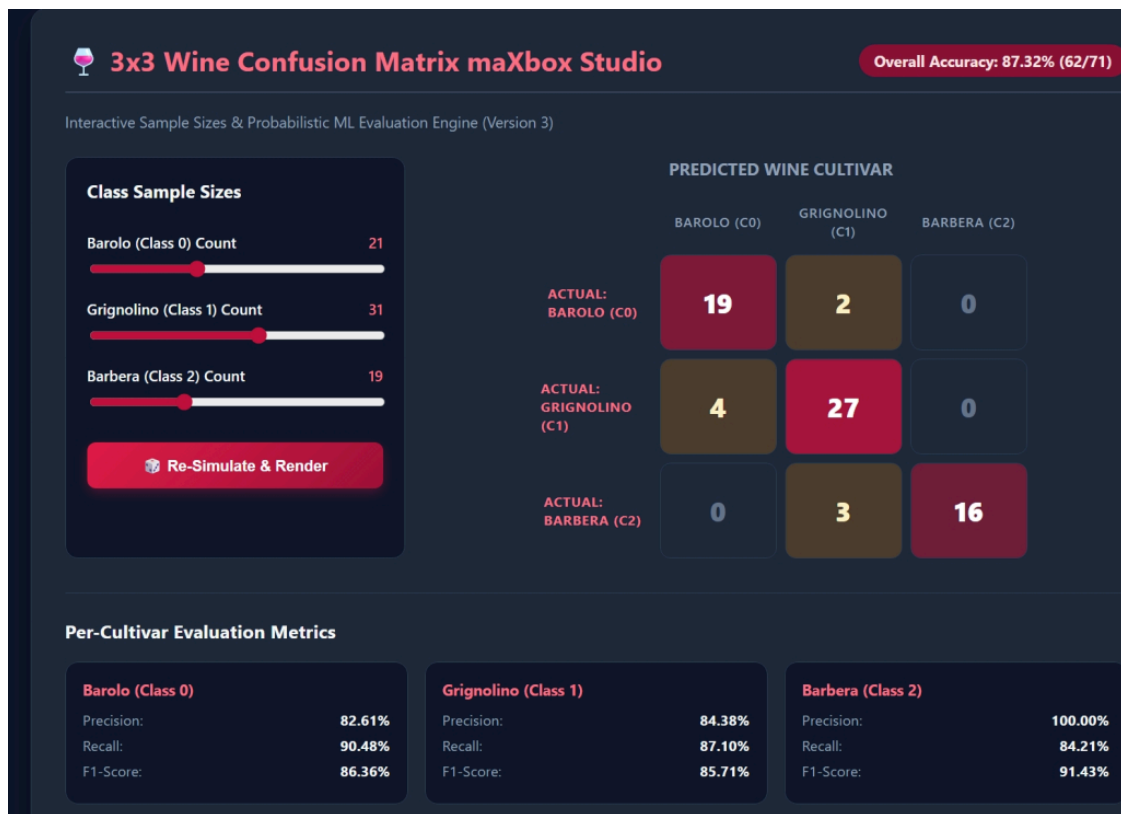
https://sourceforge.net/projects/maxbox5/files/examples/1503_native_pascal_confusion_matrix22wine_r_andtrack.pas/download

JS Web Version

Here is the complete, self-contained HTML/JavaScript Interactive Dashboard for Version 3 (3×3 Wine Dataset Confusion Matrix with dynamic sample size trackbars and randomized predictions).

Key Interactive Features

1. 3 Sample Size Sliders: Adjust sample counts for Barolo, Grignolino, and Barbera (range 5–50 samples each).
2. Re-Simulate Button: Generates new probabilistic classifier evaluations with realistic noise (~88%–92% accuracy).
3. Dynamic Heatmap Grid: Shades correct diagonal classifications in burgundy red (#be123c) and off-diagonal misclassifications in amber (#d97706).
4. Real-time Per-Class Metrics: Automatically computes Precision, Recall, and F1-Scores for all three wine cultivars alongside overall accuracy.



<https://raw.githubusercontent.com/maxkleiner/maXbox5/main/assets/machinelearning56.htm>

<https://raw.githubusercontent.com/maxkleiner/maXbox5/main/assets/machinelearning56.htm>

1. 3×3 Baseline Matrix Layout

Actual \ Predicted	Barolo (C0)	Grignolino (C1)	Barbera (C2)	Actual Total
Barolo (C0)	18 (TP)	0	0	18
Grignolino (C1)	0	20 (TP)	1 (FN to C2)	21
Barbera (C2)	0	1 (FN to C1)	14 (TP)	15
Predicted Total	18	21	15	54

https://sourceforge.net/projects/maxbox5/files/examples/1504_confusion_matrix22wine_randtrack_simulator.pas/download

2. Per-Cultivar Performance Metrics

- Barolo (Class 0)
 - Precision: 100.0% ($\frac{18}{18}$)
 - Recall: 100.0% ($\frac{18}{18}$)
 - F1-Score: 100.0%
- Grignolino (Class 1)
 - Precision: 95.24% ($\frac{20}{20 + 1}$) — 1 Barbera instance misclassified as Grignolino.

- Recall: 95.24% ($\frac{20}{21}$) — 1 Grignolino instance missed and categorized as Barbera.
- F1-Score: 95.24%
- Barbera (Class 2)
 - Precision: 93.33% ($\frac{14}{14 + 1}$) — 1 Grignolino instance misclassified as Barbera.
 - Recall: 93.33% ($\frac{14}{15}$) — 1 Barbera instance missed and categorized as Grignolino.
 - F1-Score: 93.33%

Advertisements

Occasionally, some of your visitors may see an advertisement here, as well as a [Privacy & Cookies banner](#) at the bottom of the page. You can hide ads completely by upgrading to one of our paid plans.

UPGRADE NOW

DISMISS MESSAGE

Posted in [Code](#), [Geo Intelligence](#), [machine learning](#), [maXbox](#), [Statistics](#)

Leave a Reply

Write a comment...



maxkleiner [Log out](#)

Comment

← [From Bayes Theorem to Confusion Matrix](#)

[Create a free website or blog at WordPress.com.](#)