



CS 6350/DS 4350: HW1 – Q1- Classification Metrics (Binary Case)

Samir Abdelrahman

Yujin Song, Shubham Sawant

Fall 2025



Grading Rubric

Question	Points	Scoring rules	Graded Points
1-(a) Precision	12	Formula - 4 pts Test case - using counts from confusion.csv (TP=95, FP=10) - 4 pts Final value - 4 pts	
1-(a) Recall	12	Formula - 4 pts Test case - using counts from confusion.csv (TP=95, FN=5) - 4 pts Final value - 4 pts	
1-(a) F1-score	12	Formula - 4 pts Test case - using the calculated P and R from above questions- 4 pts Final value - 4 pts	
1-(a) Specificity	7	Formula - 3 pts Test case - using counts from confusion.csv (TN=90, FP=10) - 2 pts Final value - 2 pts	
1-(a) Accuracy	7	Formula - 3 pts Test case - using counts from confusion.csv (TP=95, TN=90, Total=200) - 2 pts Final value - 2 pts	
Total:	50		

Use counts from confusion.csv

If a denominator is zero, award full credit for safe-divide behavior (return 0.0)

BONUS:

1-(b) AUC (Trapezoid)	10	ROC points derived correctly - 5 pts Final trapezoid area - 5 pts	
1-(b) AUC (Rank)	10	Rank/Mann–Whitney method applied - 5 pts Final value - 5 pts	
Total:	20		

Penalties: up to -10 total.

Missing work/formulas: (-3)

wrong rounding beyond ± 0.0005 : (-1)

using sklearn: (-4)

hard-coding answers without derivation: (-2).



Bonus scope

- ROC/AUC is a BONUS task (+20 points) on top of the base Q1 score (50 points). You can earn full base credit without the bonus. The bonus only applies if your code follows the rules below.

Implement from scratch requirement

- Do not use **any programming language library function** that directly computes a metric (precision, recall, F1, specificity, accuracy, confusion matrix, ROC curve, AUC, etc.).
- You may use general-purpose math/array utilities to implement the logic yourself (e.g., NumPy operations, `numpy.trapz` for trapezoidal integration).

What you may use

- numpy, pandas
- Standard library: math, csv, json, etc.
- Not allowed: scikit-learn, statsmodels, torchmetrics, tensorflow/keras metrics, xgboost/lightgbm built-in metrics, or any package function whose purpose is to compute a metric for you

Hidden Test Cases

- We will run additional hidden cases. If your code raises runtime errors on any official case (e.g., `ZeroDivisionError`, `IndexError`, `ValueError`), the grade for Q1 will be deducted.



Problem 1

In this exercise, you will compute key binary classification metrics from scratch, without using sklearn or any external ML libraries.

Data Files:

1) confusion.csv (2×2 matrix; rows = ACTUAL, columns = PREDICTED)

TN,FP

FN,TP

Example in this assignment:

90,10

5,95

=> TN=90, FP=10, FN=5, TP=95.

2) scores.csv (two columns, NO header):

score,label

- score (float): larger = more 'positive' and it is labeled 1 when above a chosen threshold
- label (int): ground-truth class in {0,1}

Questions:

- (a) [50 pts] Using confusion.csv, compute: Precision, Recall, F1-score, Specificity, Accuracy.
- (b) [Bonus] [20 pts] Using scores.csv, plot or tabulate the ROC curve points, and compute the AUC using:
 - trapezoidal rule,
 - rank (Mann–Whitney) method.
- (c) [Bonus / included in above 20] Report all results clearly.