Logistic Regression Metrics Explained with Example

✅ Quick Recap: Logistic Regression
It predicts probabilities and classifies data into two categories (e.g., 0 or 1).

🧠 Let's Say...
You're predicting if a person has heart disease using Logistic Regression.

1 = Has Heart Disease (Positive)
0 = No Heart Disease (Negative)

📊 Sample Confusion Matrix

	Predicted: 1	Predicted: 0
Actual: 1 (Positive)	TP = 80	FN = 20
Actual: 0 (Negative)	FP = 10	TN = 90

This means:
- 80 people who had disease were correctly predicted (True Positive)
- 20 people who had disease were missed (False Negative)
- 10 people without disease were wrongly predicted as sick (False Positive)
- 90 healthy people correctly predicted (True Negative)

1. Accuracy

📌 How many total predictions are correct?

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy = (80 + 90) / (80 + 90 + 10 + 20) = 170 / 200 = 85%

✅ Use when: Classes are balanced.

2. Precision

📌 Out of all predicted positive cases, how many were truly positive?

Precision = TP / (TP + FP)

Precision = 80 / (80 + 10) = 80 / 90 ≈ 88.9%

✅ Use when: False Positives are costly (e.g., fraud detection, medical diagnosis).

3. Recall (Sensitivity / True Positive Rate)

📌 Out of all actual positive cases, how many did the model catch?

Recall = TP / (TP + FN)

Recall = 80 / (80 + 20) = 80 / 100 = 80%

✅ Use when: False Negatives are dangerous (e.g., cancer detection, missed disease).

4. F1 Score

📌 Harmonic mean of Precision and Recall

F1 Score = 2 × (Precision × Recall) / (Precision + Recall)

F1 Score = 2 × (0.889 × 0.80) / (0.889 + 0.80) ≈ 0.842

✅ Use when: Need balance between precision and recall.

📌 Summary Table

Metric	Tells You...	When to Use
Accuracy	Overall correctness	Balanced classes
Precision	How often positive predictions were correct	When False Positives are costly
Recall	How many actual positives were caught	When False Negatives are risky
F1 Score	Balance between precision & recall	When both FP and FN matter
Confusion Matrix	Complete picture of prediction performance	For deep error analysis