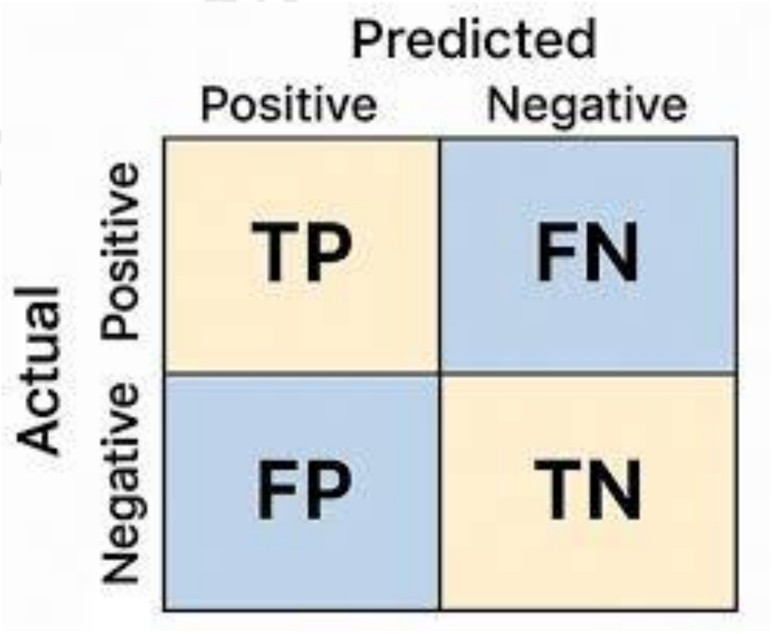
CSC 36000:
Modern Distributed Computing *NextGen with AI Agents*

By Saptarashmi Bandyopadhyay
Email: sbandyopadhyay@ccny.cuny.edu
Assistant Professor of Computer Science
City College of New York and Graduate Center at City University of New York

Evaluating Model Performance in Distributed Computing and Decentralized AI

Confusion Matrix

- Visualizes performance of a classification model
- What kinds of errors does the model make? When is it the most accurate?
- Key Terms:
  - **True Positive**
    - Predicted True, Actual True
  - **False Positive (Type 1 Error)**
    - Predicted True, Actual False
  - **True Negative**
    - Predicted False, Actual False
  - **False Negative (Type 2 Error)**
    - Predicted False, Actual True
- Accuracy, Precision, Recall, and F-measure can be traced back to this!



Accuracy

- Accuracy Measures the percentage of predictions that are correct
- This is most useful when classes (positives vs negatives) are relatively balanced
- Warning: Can be misleading with imbalanced classes

$$\frac{TP + FN}{TP + TN + FP + FN}$$

Precision

- The proportion of positively identified examples
- Precision can be thought of as the percentage of the “caught” items that are actually in the positive class, as opposed to being mistaken

$$\frac{TP}{TP + FP}$$

Example: Precision in Distributed GPU Clusters

- When training an AI model across multiple GPUs in a cluster, different precision scores across different GPUs could signify an imbalance in your data distribution
- By tracking precision, you can make sure that the learning process is going smoothly across all GPUs
- If some GPUs have low precision, there may be problems with your model parallelization techniques

Recall

- The proportion of positives that were identified correctly out of all the positive examples
- Most important when the "cost" of a false negative is high, e.g. misdiagnosing a patient

$$\frac{TP}{TP + FN}$$

Example: Self-Driving Cars

- Because recall identifies what what ID'd correctly out of the total number of "hits", it's very useful for self driving cars
- Recall can track the total number of pedestrians and other hazards identified
- It can also be used to monitor important signals like stop signs, traffic lights, and speed limits
- Even slight mistakes could spell disaster!

F-measure

- The F-Measure is the harmonic mean between the precision and the recall
- It punishes an imbalance between the precision and the recall
- It’s most useful when the number of positives and the number of negatives is lopsided

$$2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Why the harmonic mean?

- Imagine your model gets 99% of positive examples correct and 1% of negative examples correct
- If you were to take the arithmetic mean (aka the accuracy), you would get 50%
- This doesn't consider at all the failure to correctly identify all examples
- This same model would have an F-measure of ~2%, clearly letting you know the model needs debugging

Example: Monitoring Autonomous Systems

- F-measure can be used when both precision *and* recall are important
- If you're operating a computing cluster, it's important to both:
    - Identify malfunctioning nodes accurately (high recall)
    - Correctly flag faulty nodes (high precision)
- More applications:
    - Intelligent job prioritization (e.g. SLURM clusters)
    - Autonomous Management Systems

When do I use each metric?

- Start with the **confusion matrix**
  - This will give you solid footing for exploring the other metrics and understanding what kinds of errors your model is making
- Use **accuracy** when you want a quick glance at how the errors across your (balanced) dataset are distributed
- Choose **precision** when avoiding False Positives is important, and **recall** when False Negatives are important
- **F-measure** is most useful as a reliable metric for unbalanced positives vs negatives in your dataset

Questions?