CS 260: Foundations of Data Science

Prof. Thao Nguyen

Fall 2024

Materials by Sara Mathieson

Admin

- **Lab 2** grades & feedback will be posted on Wednesday
- **Lab 3** due tonight
- **Lab 4** posted, due next Monday at midnight
- **Lecture Notes**

Peer Tutoring

- **Student tutors** (Fejiro Anigbro, Darshan Mehta)
- **Flexible hours**
- **Free!**

TECH TALKS 2024

OCTOBER 7,8 & 9TH | 6-8PM EST

Sign up for a 30 minute virtual informational interview with a Tri-Co alum to gain tech career insights!

Alumni will represent various tech roles including software engineering and development, data science, tech consulting, product management and biotech.

OCT 7	OCT 8	OCT 9
Accenture	Bristol Myers Squibb	The Walt Disney Company
FERMAT Commerce	Community.com	Fresh Tracks Insights
Grubhub	C3 Presents (Live Nation)	Meta
	Opower (Oracle)	Grubhub

TRI-COLLEGE RECRUITING CONSORTIUM

HAVERFORD | BRYN MAWR | SWARTHMORE

Outline for today

- Recap SGD (stochastic gradient descent)

- Introduction to classification
  - Decision tree models
  - Probabilistic interpretation

- Evaluation Metrics
  - Confusion matrices
  - Precision and recall
  - ROC curves

Outline for today

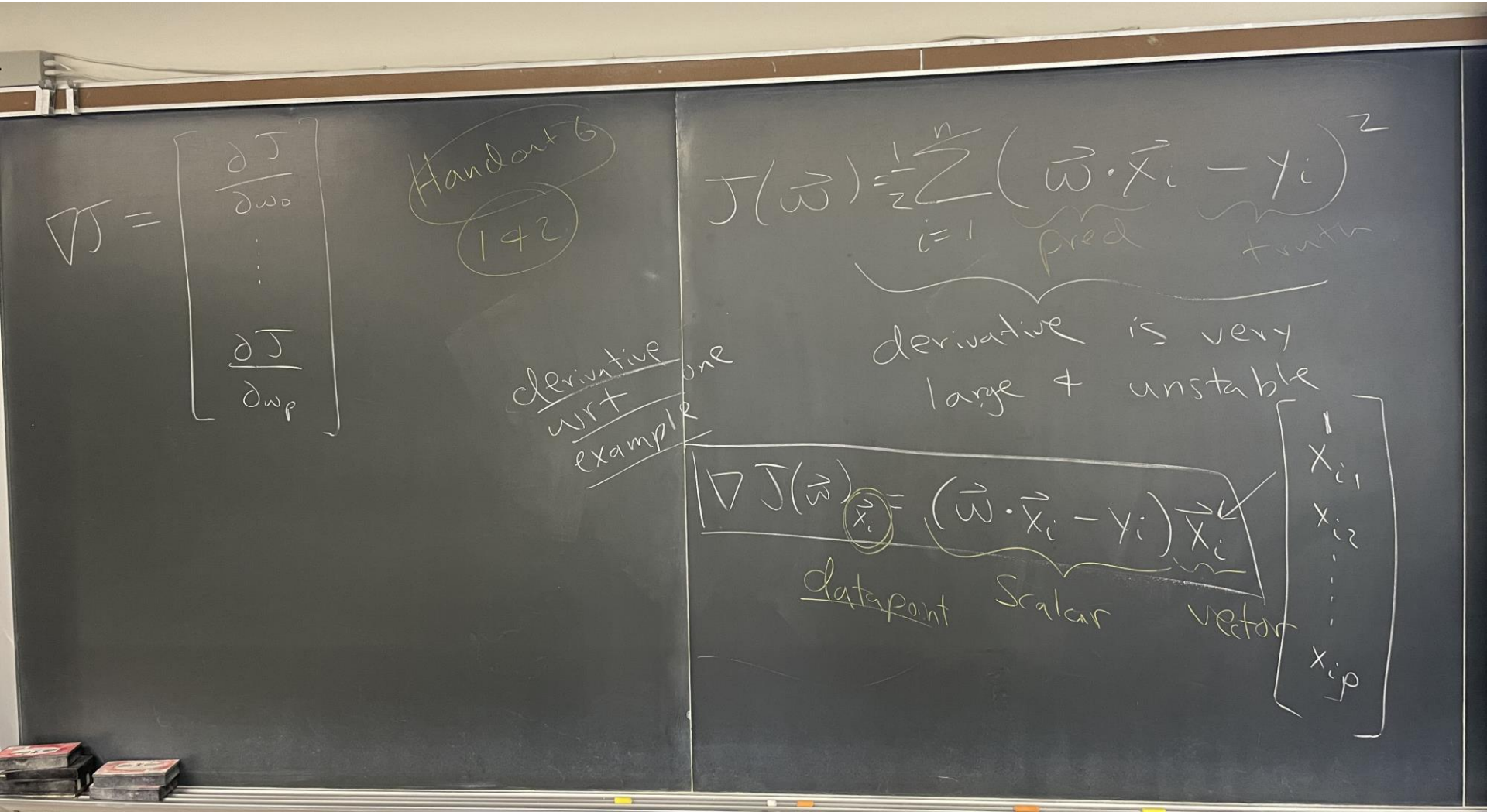
- Recap SGD (stochastic gradient descent)

- Introduction to classification
  - Decision tree models
  - Probabilistic interpretation

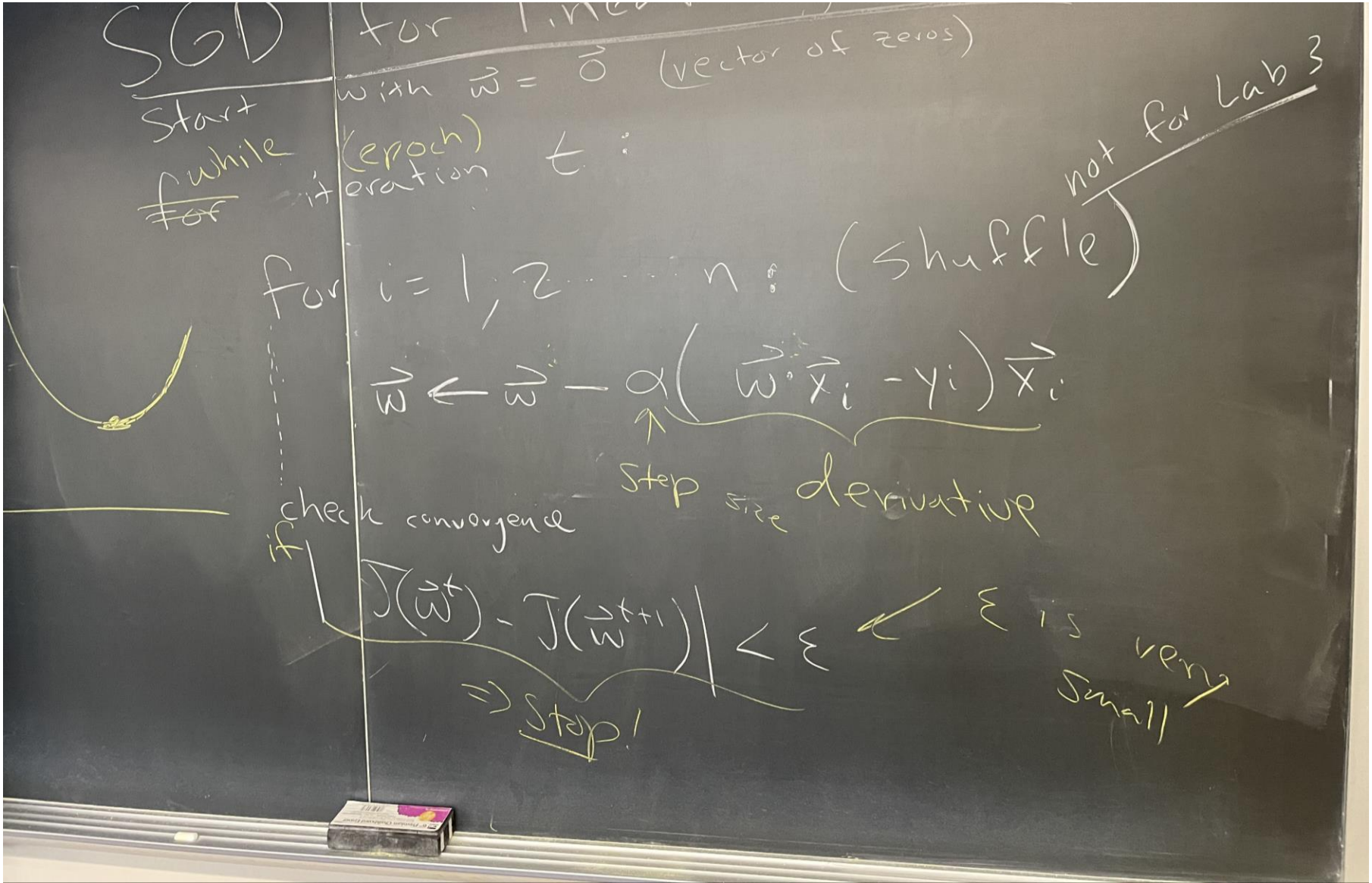
- Evaluation Metrics
  - Confusion matrices
  - Precision and recall
  - ROC curves

Stochastic Gradient Descent for Linear Regression

Key Idea: take the derivative of **one datapoint** at a time and use that to update w



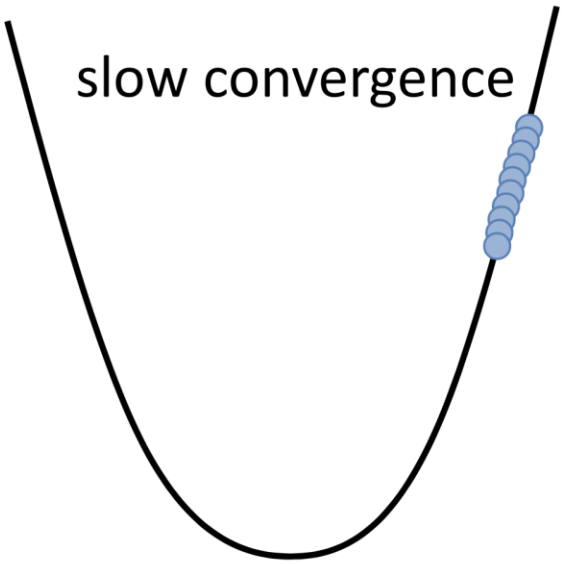
Stochastic Gradient Descent for Linear Regression



Choosing the step size alpha

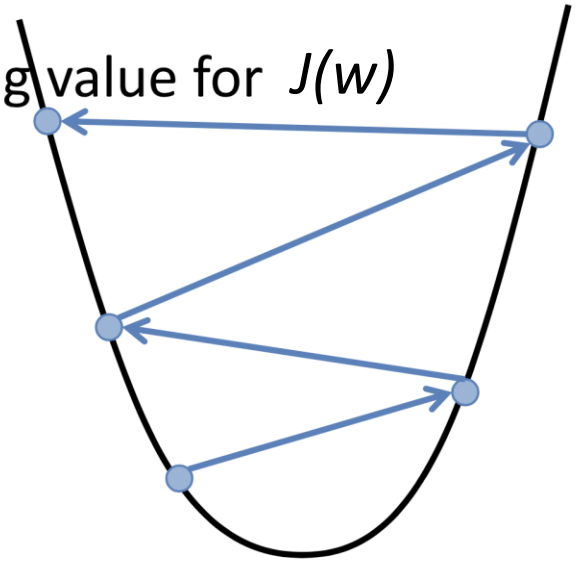
α too small

slow convergence



α too large

increasing value for $J(w)$



- may overshoot minimum
- may fail to converge (may even diverge)

Slide: modified from Jessica Wu
Based on slide by Eric Eaton
[Originally by Andrew Ng]

Pros and Cons

Gradient Descent

- requires multiple iterations
- need to choose α
- works well when p is large
- can support online learning

(Analytic Solution)

Normal Equations

- non-iterative
- no need for α
- slow if p is large
  - matrix inversion is $O(p^3)$

Slide: modified from Jessica Wu
Based on slide by Eric Eaton

Outline for today

- Recap SGD (stochastic gradient descent)

- Introduction to classification
  - Decision tree models
  - Probabilistic interpretation

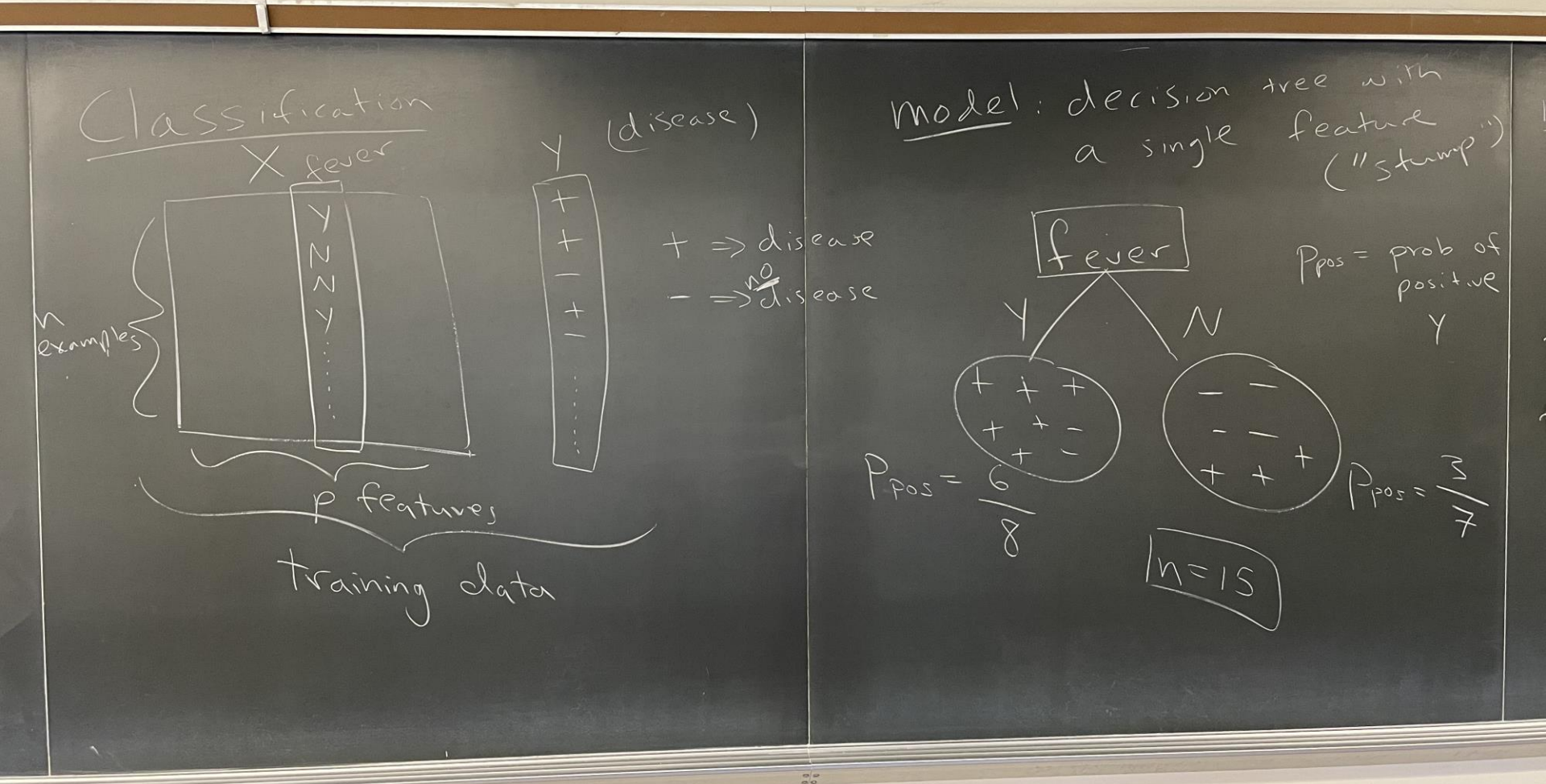
- Evaluation Metrics
  - Confusion matrices
  - Precision and recall
  - ROC curves

Binary classification examples

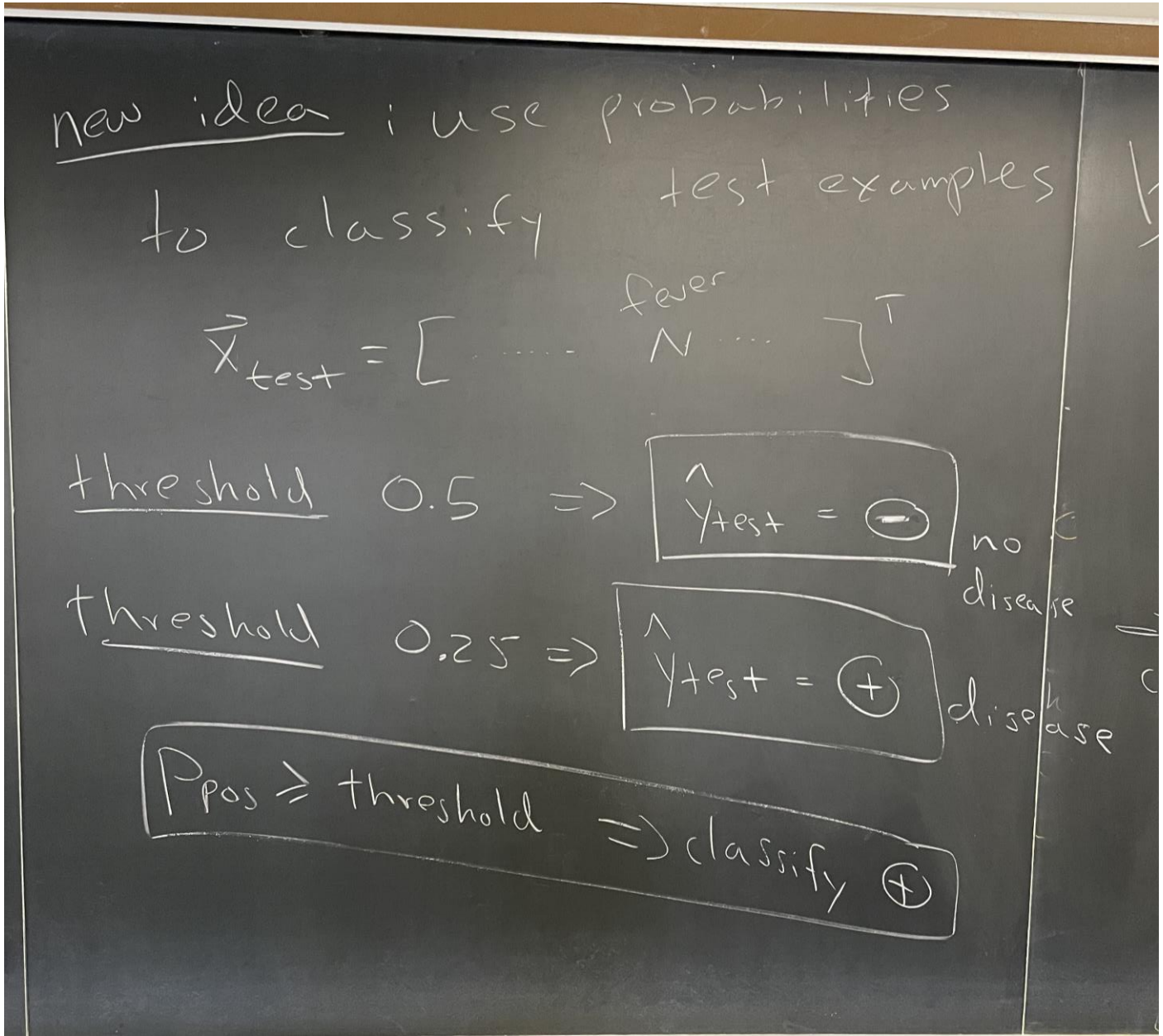
- Transactions that indicate credit card fraud
- Accounts that are bots
- Detecting which scans show tumors
- Prenatal test for Down's Syndrome
- Finding genes under natural selection
- Regions of the environment that contains the object the robot is searching for

In all these examples, we are trying to find unusual items ("needle in a haystack") -- we call these *positives*

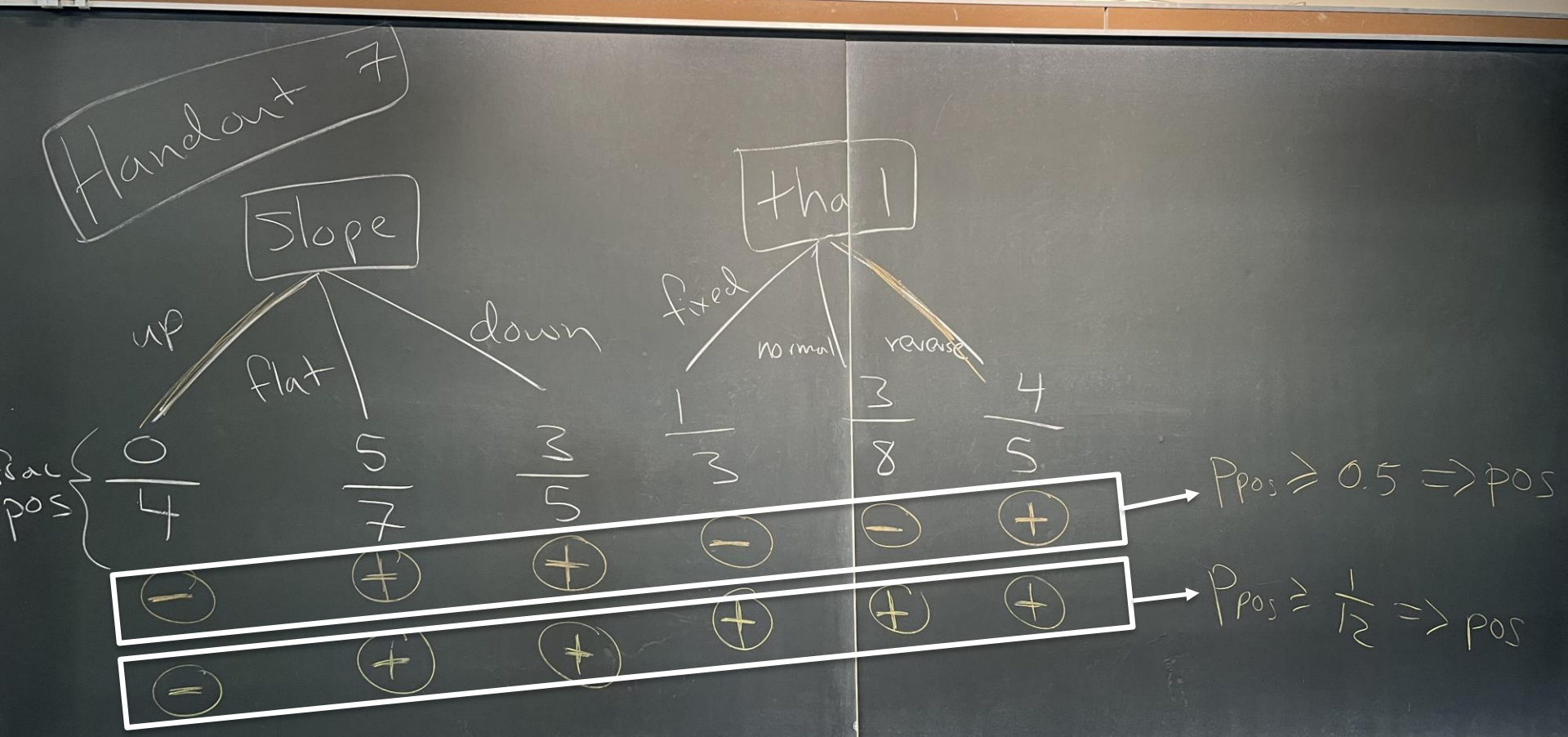
Introduction to Classification



Introduction to Classification



Handout 7


Handout 7
Slope
up
flat
down
0/4
5/7
3/5
thal
fixed
normal
reverse
1/3
3/8
4/5
pos
$P_{pos} \geq 0.5 \Rightarrow$ pos
$P_{pos} \geq \frac{1}{2} \Rightarrow$ pos

Outline for today

- Recap SGD (stochastic gradient descent)

- Introduction to classification
  - Decision tree models
  - Probabilistic interpretation

- Evaluation Metrics
  - Confusion matrices
  - Precision and recall
  - ROC curves

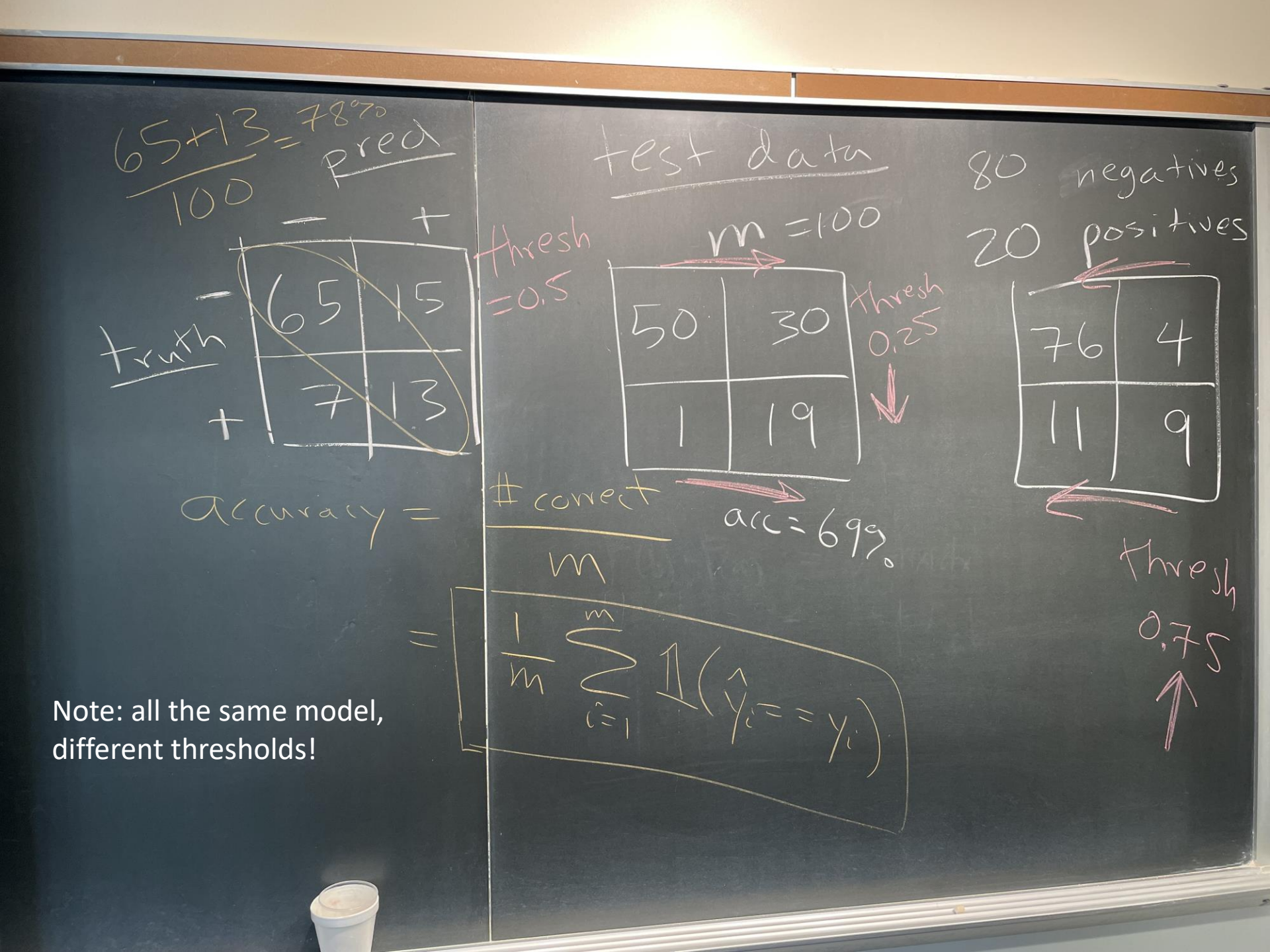
Goals of Evaluation

- Think about what metrics are important for the problem at hand
- Compare different methods or models on the same problem
- Common set of tools that other researchers/users can understand

Training and Testing

(high-level idea)

- **Separate** data into "**train**" and "**test**"
  - *n* = num training examples
  - *m* = num testing examples
- **Fit** (create) the model using **training data**
  - e.g. sea_ice_1979-2012.csv
- **Evaluate** the model using **testing data**
  - e.g. sea_ice_2013-2020.csv

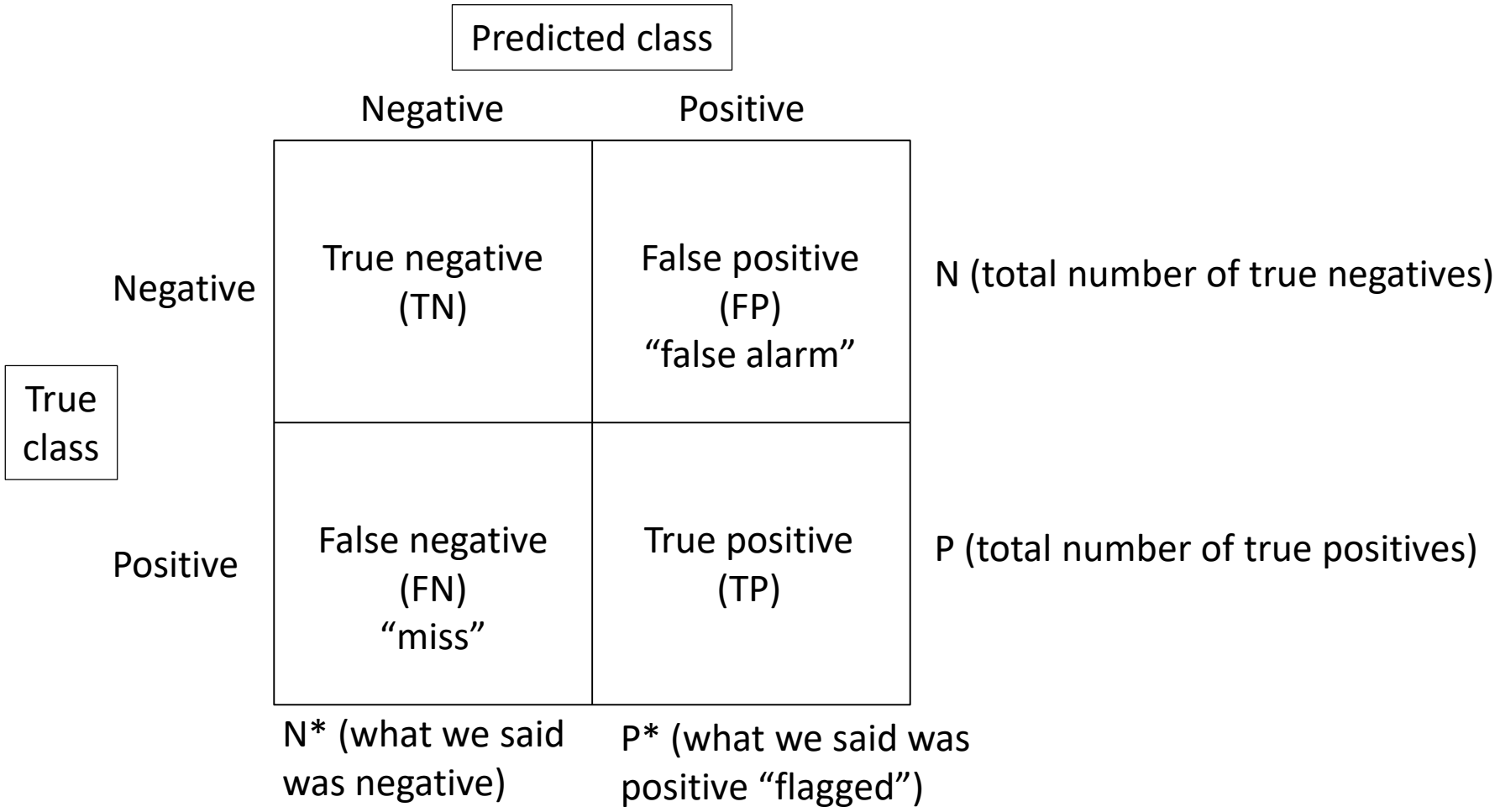


Note: all the same model, different thresholds!

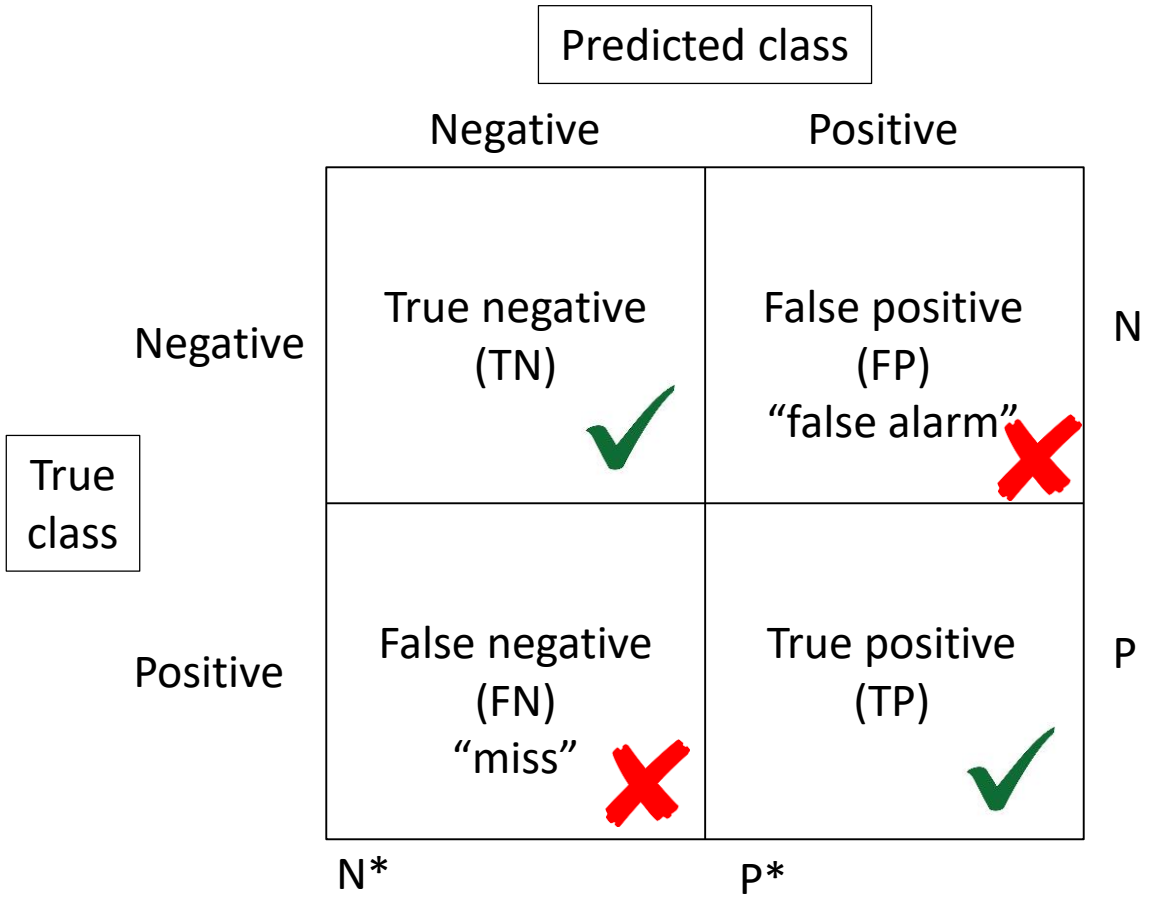
Confusion Matrices

True class \ Predicted class	Negative	Positive
Negative	True negative (TN)	False positive (FP)
Positive	False negative (FN)	True positive (TP)

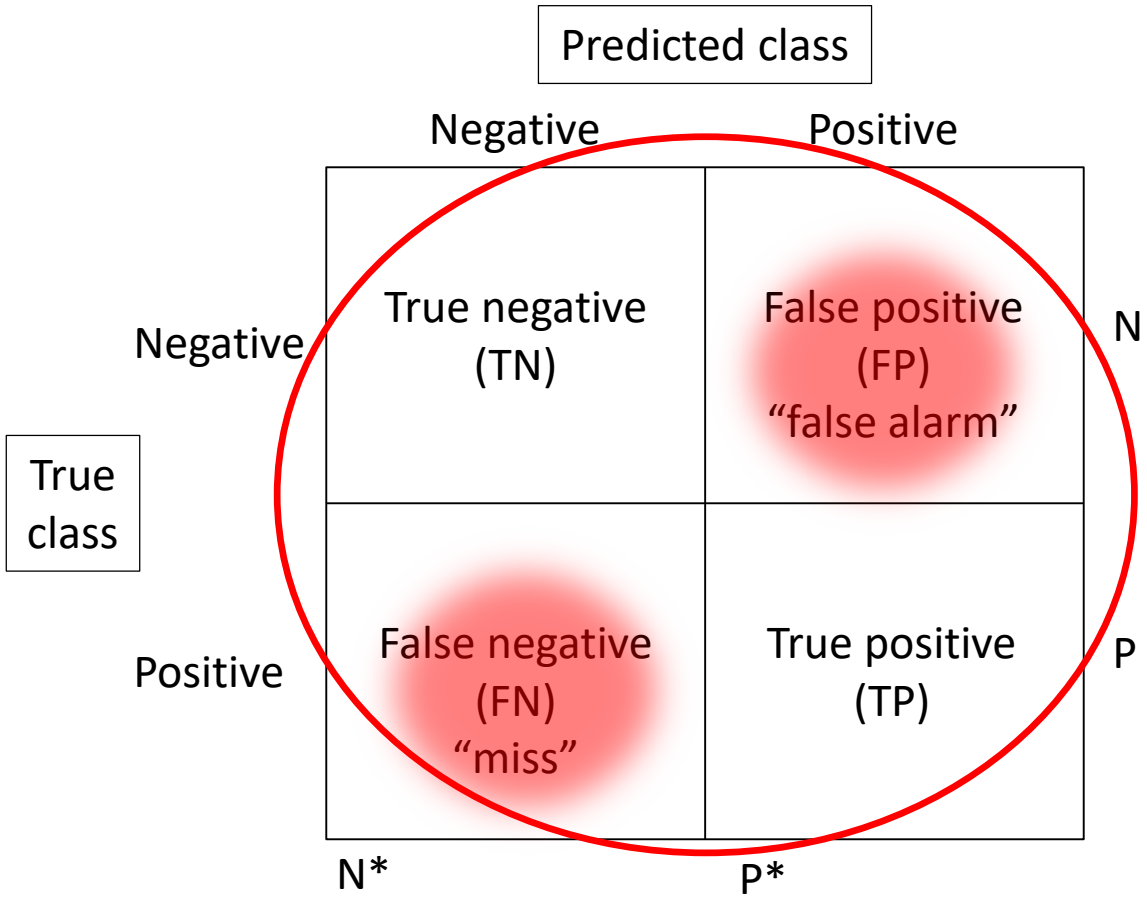
Confusion Matrices



Confusion Matrices



Confusion Matrices

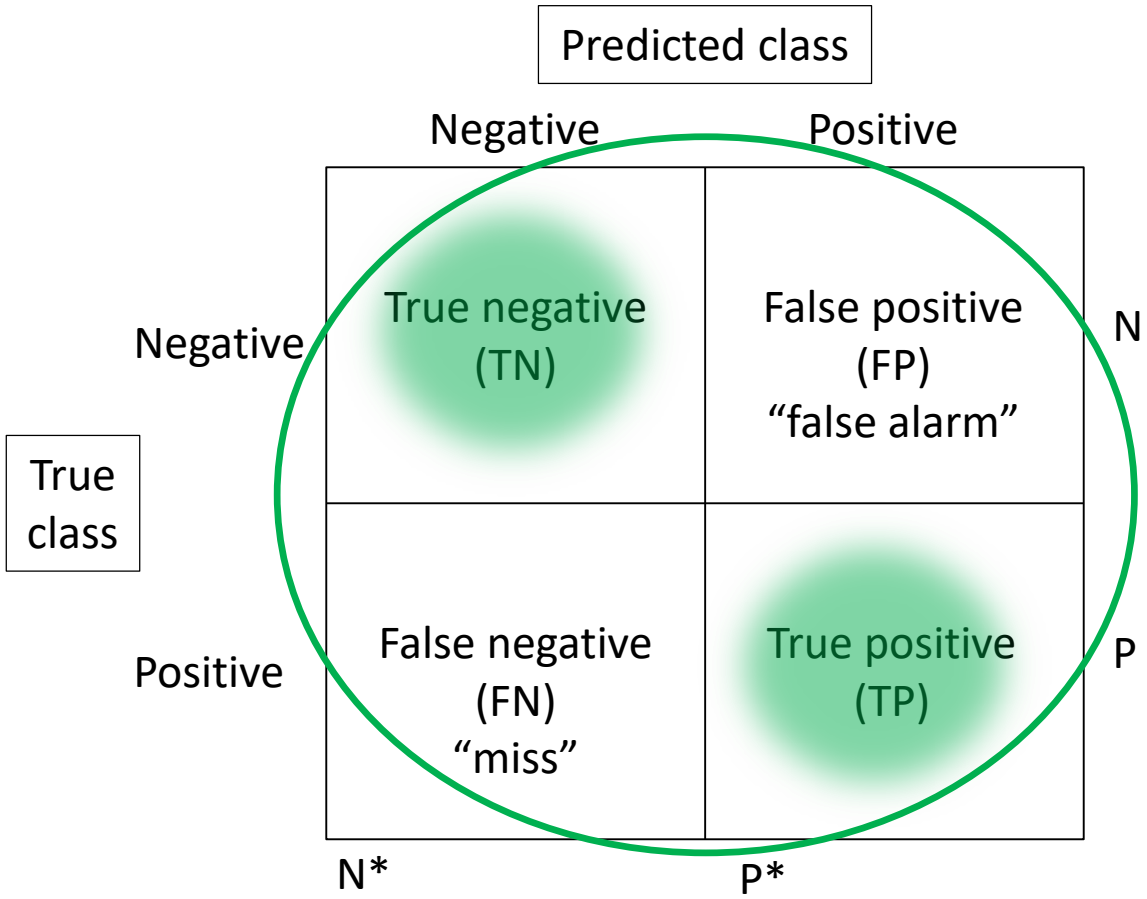


Error:

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

= (FN+FP)/(N+P)

Confusion Matrices

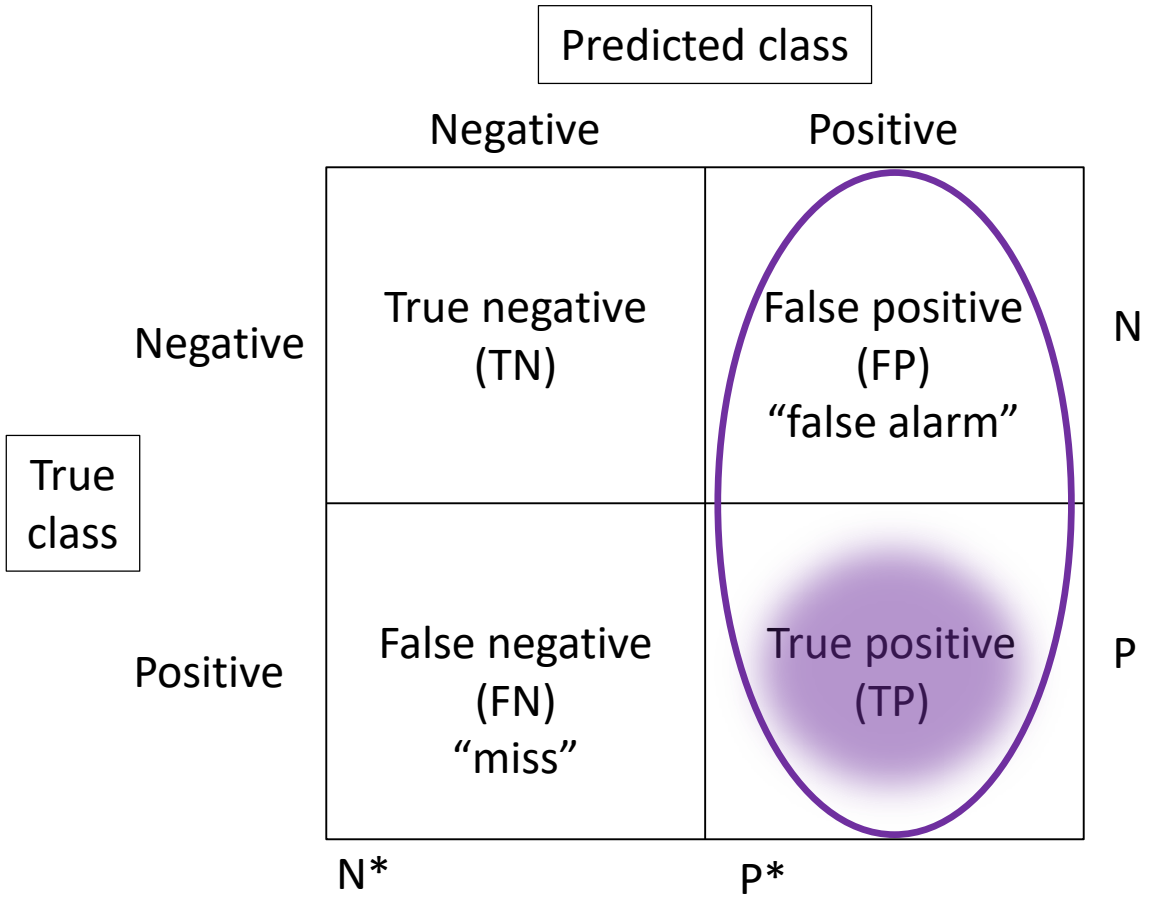


Accuracy = 1-Error:

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

= (TN+TP)/(N+P)

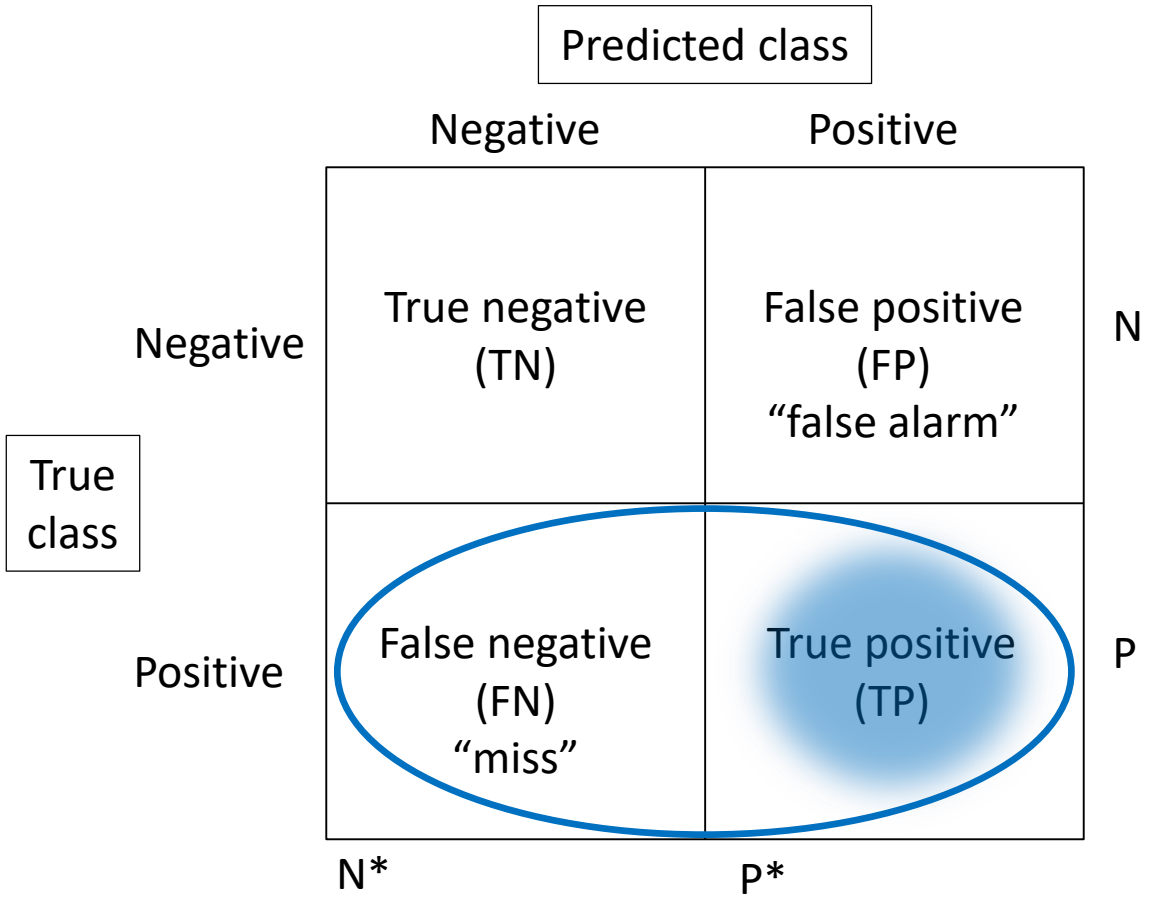
Confusion Matrices



Precision:

TP/(FP+TP) = TP/P*

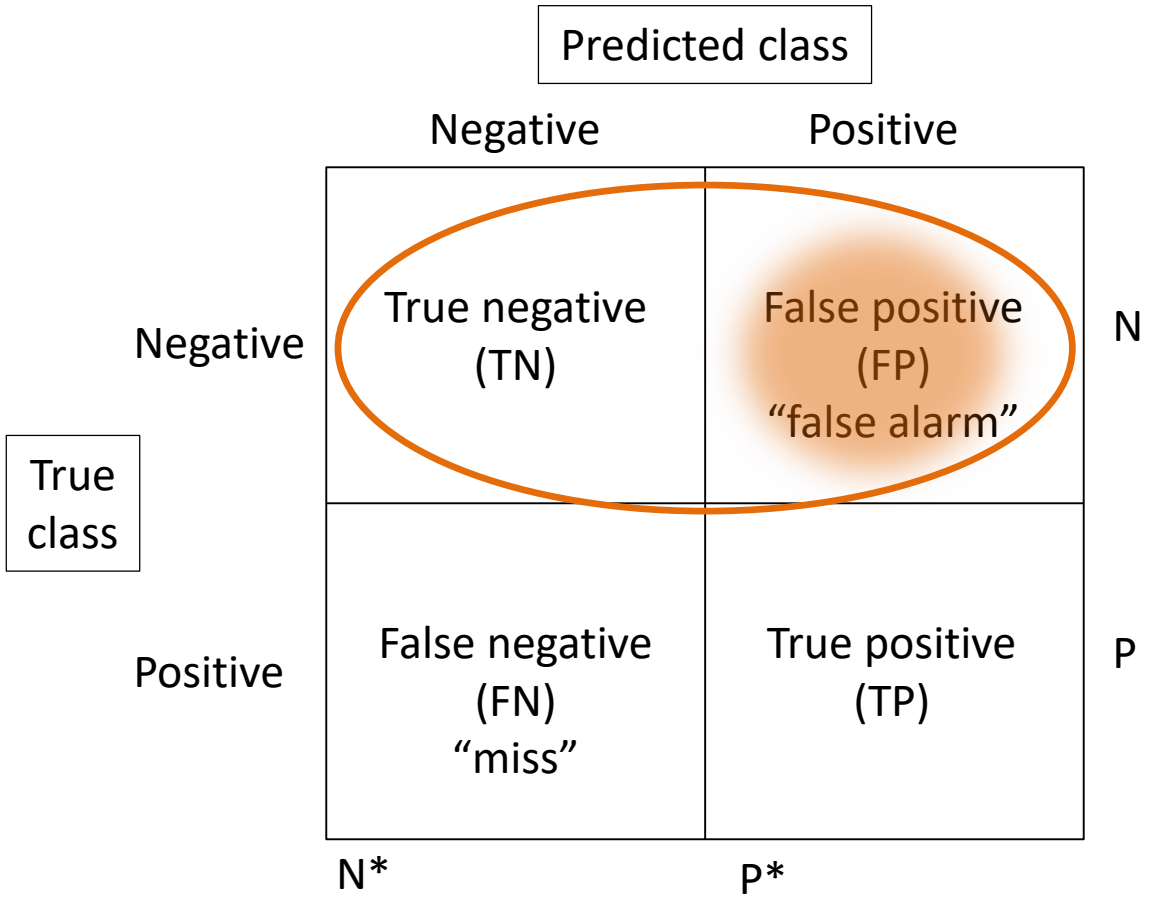
Confusion Matrices



Recall
(True Positive Rate):

TP/(FN+TP) = TP/P

Confusion Matrices



False Positive Rate:

FP/(TN+FP) = FP/N

Precision and Recall

- Precision: of all the "flagged" examples, which ones are actually relevant (i.e. positive)?

  (Purity)

- Recall: of all the relevant results, which ones did I actually return?

  (Completeness)