

# CS 260: Foundations of Data Science

Prof. Sara Mathieson

Spring 2025



**Haverford**  
COLLEGE

# Admin

- **Lab 1** grades posted on Moodle
  - Note: binary search is  $O(\log n)$
- **Lab 3** due tomorrow night
  - I'll check in with everyone today about Lab 3 during lab
- **Lab 4** posted (pair-programming required, different partner)

# Stochastic Gradient Descent (high-level)

```
set  $\mathbf{w} = \mathbf{0}$  vector
while cost  $J(\mathbf{w})$  still changing (or max iter reached):
    shuffle data points
    for  $i = 1 \dots n$ :
         $\mathbf{w} \leftarrow \mathbf{w} - \text{alpha}(\text{derivative of } J(\mathbf{w}) \text{ wrt } x_i)$ 
    store  $J(\mathbf{w})$ 
```

# Mini-quiz for linear regression

For each of the following terms/descriptions, write out the corresponding equation:

- 1) Linear regression **model**
- 2) Linear regression **cost function**
- 3) **Gradient** of cost function wrt one datapoint
- 4) **Gradient descent** weight vector update



# Mini-quiz for linear regression

Linear Regression

$$\textcircled{1} \quad \hat{y} = \vec{w} \cdot \vec{x}$$

$$\textcircled{2} \quad J(\vec{w}) = \frac{1}{2} \sum_{i=1}^n (\underbrace{y_i}_{\text{truth}} - \underbrace{\vec{w} \cdot \vec{x}_i}_{\text{pred}})^2$$

$$\textcircled{3} \quad \nabla_{\vec{x}_i} J(\vec{w}) = (\vec{w} \cdot \vec{x}_i - y_i) \vec{x}_i$$

"derivative"  $\nearrow$   $\searrow$

$$\textcircled{4} \quad \vec{w} \leftarrow \vec{w} - \alpha (\nabla_{\vec{x}_i} J(\vec{w}))$$

# Outline

- Introduction to classification
  - Decision tree models
  - Probabilistic interpretation
- Evaluation Metrics
  - Confusion matrices
  - Precision and recall
  - ROC curves

# Outline

- 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
- Finding regions of the genome with high recombination rate (“hotspots”)

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

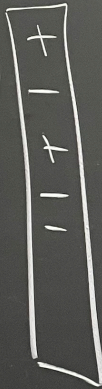
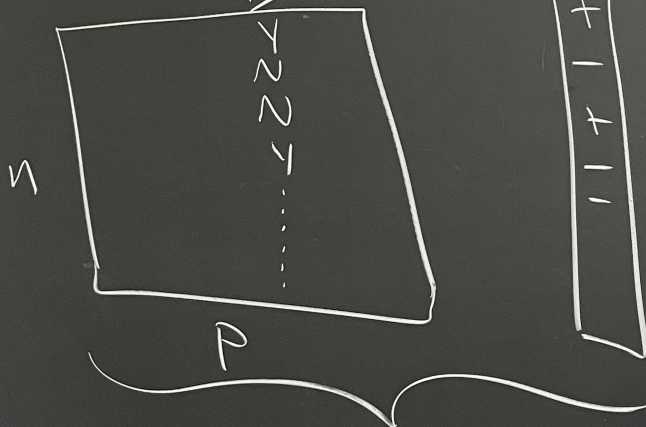


# Introduction to Classification

## Introduction to Classification

single feature models  $\vec{y}$

$x_{\text{fever}}$

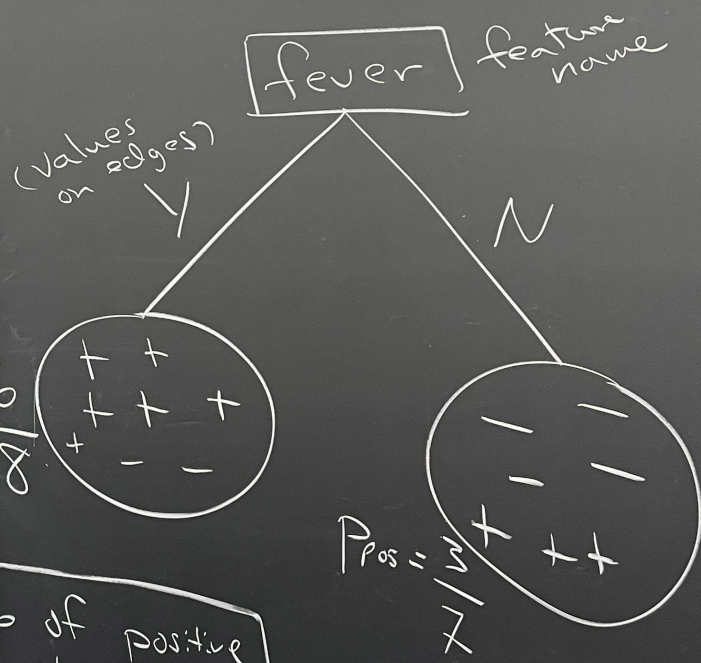


+ disease  
- no disease

$$P_{\text{pos}} = \frac{6}{8}$$

$P_{\text{pos}} = \text{prob of positive classification}$

model: decision tree with 1 feature ("stumps")





# Introduction to Classification

new idea: use probabilities  
to classify test example

$$\vec{X}_{\text{test}} = [\dots \overset{\text{fever}}{N} \dots]^T$$

$$\text{threshold} = 0.5$$

$$\hat{Y}_{\text{test}} = -$$

(no disease)

$$\text{threshold} = 0.25$$

$$\hat{Y}_{\text{test}} = +$$

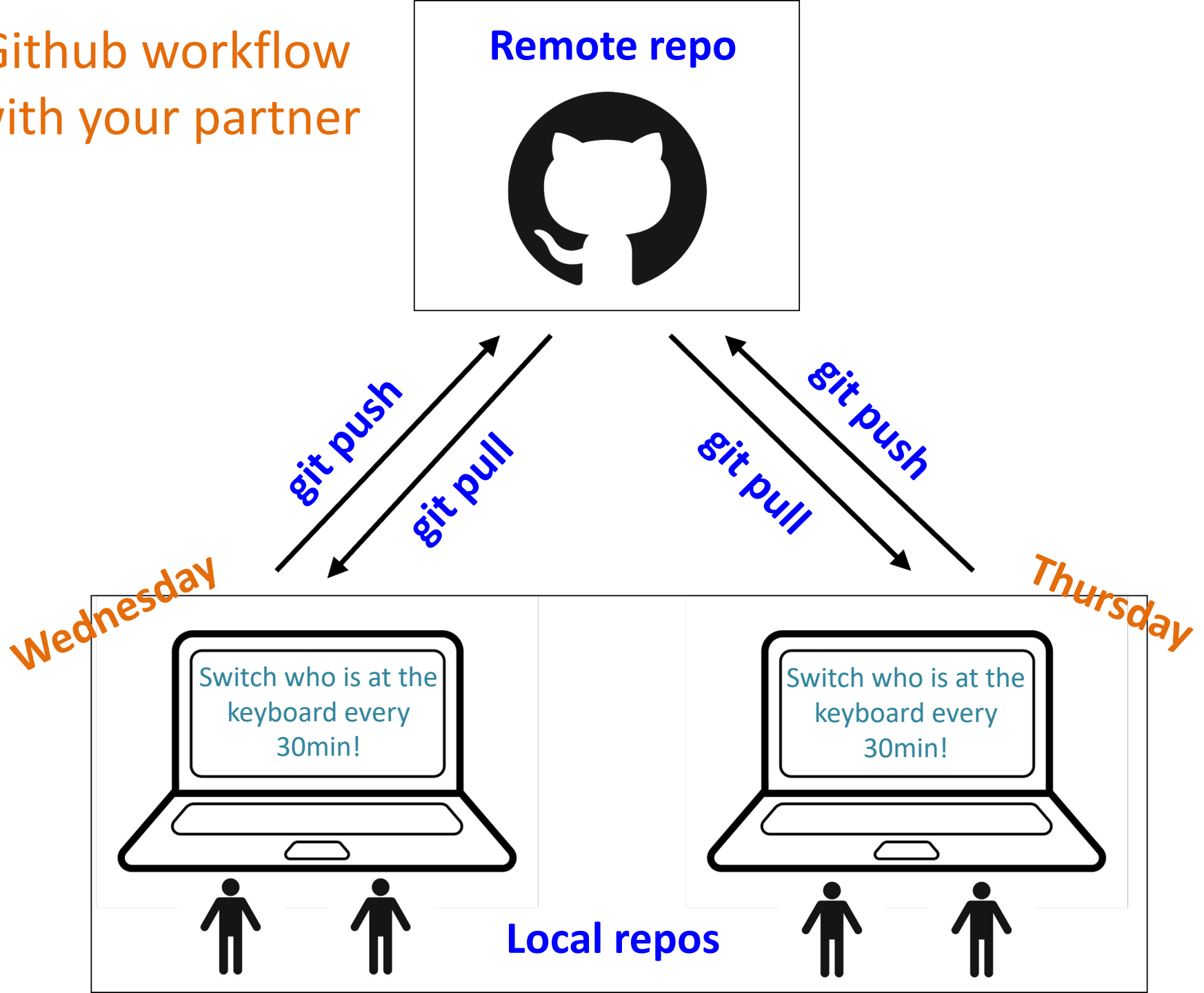
disease

# Handout 7: work with Lab 4 partner

Lab A

Lab B

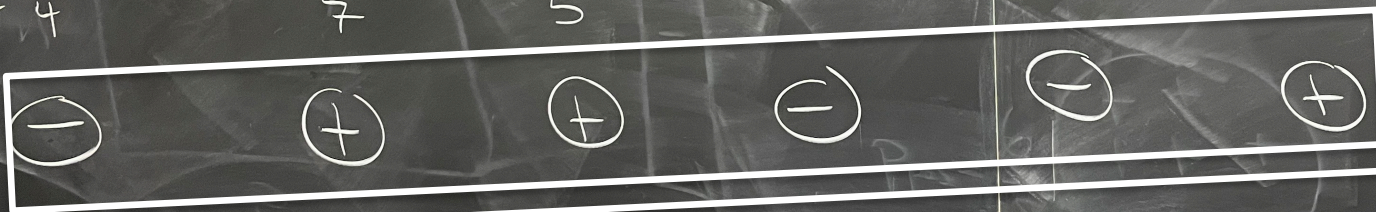
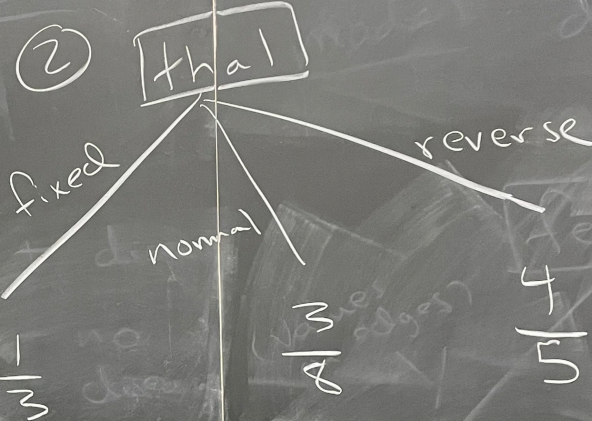
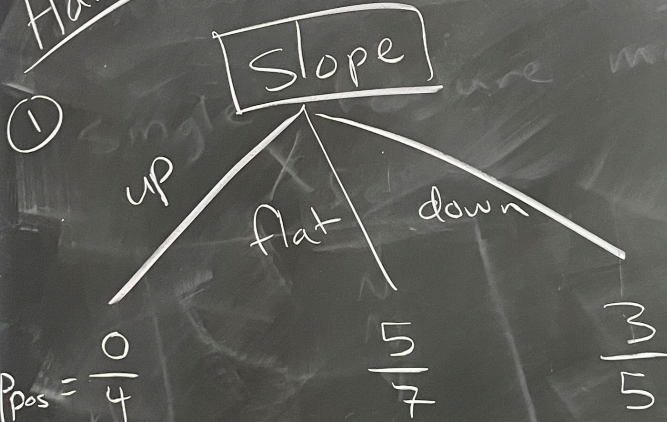
# Github workflow with your partner





# Handout 7

Handout 7



# Outline

- 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

# Confusion matrices

accuracy: (classification)

$$acc = \frac{\# \text{ correct}}{m} = \frac{1}{m} \sum_{i=1}^m \mathbb{1}(y_i = \hat{y}_i)$$

Confusion matrix

test data.

pred

(A)

	-	+	
-	65	15	→ 80
+	7	13	→ 20

(B)

	-	+	
-	50	30	→ 80
+	1	19	→ 20

(C)

	-	+	
-	76	4	
+	11	9	

1 4 0 if not matching  
negatives, 20 positives

m = 100

Note: all the same model, different thresholds!

$$\frac{65 + 13}{100} = \frac{78}{100}$$

acc = .69  
low thresh

high thresh

# Confusion Matrices

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



# Confusion Matrices

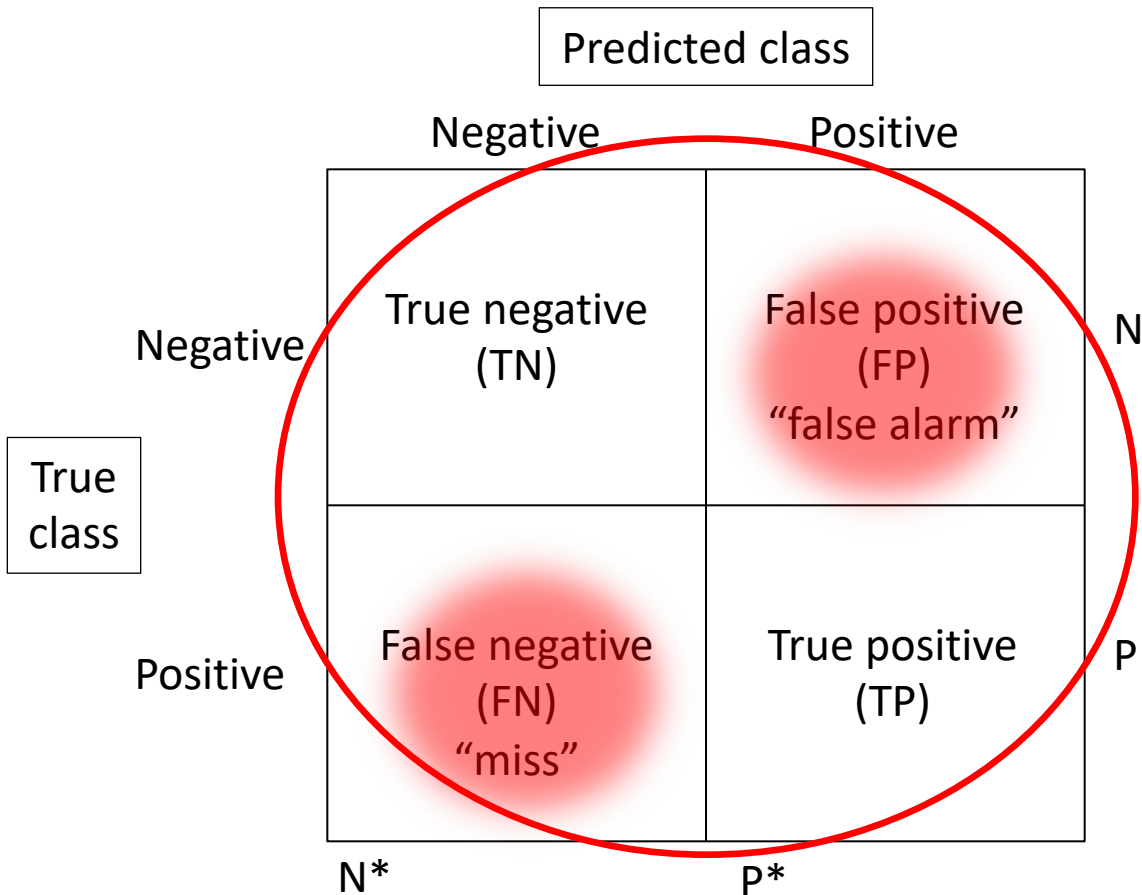
		Predicted class		
		Negative	Positive	
True class	Negative	True negative (TN)	False positive (FP) “false alarm”	N (total number of true negatives)
	Positive	False negative (FN) “miss”	True positive (TP)	P (total number of true positives)
		N* (what we said was negative)	P* (what we said was positive “flagged”)	

# Confusion Matrices

		Predicted class		
		Negative	Positive	
True class	Negative	True negative (TN) ✓	False positive (FP) "false alarm" ✗	N
	Positive	False negative (FN) "miss" ✗	True positive (TP) ✓	P
		N*	p*	



# Confusion Matrices

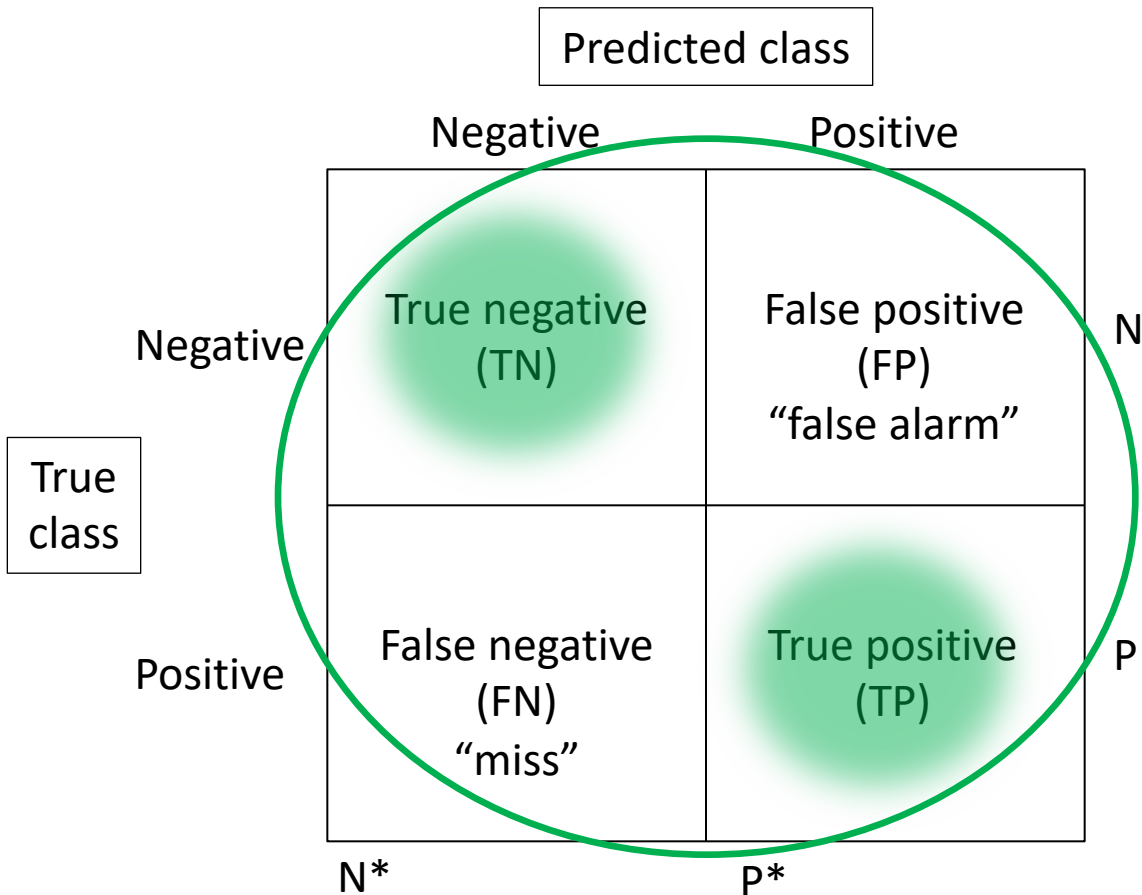


Error:

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

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

# Confusion Matrices



Accuracy = 1-Error:

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

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

# Confusion Matrices

		Predicted class		
		Negative	Positive	
True class	Negative	True negative (TN)	False positive (FP) "false alarm"	N
	Positive	False negative (FN) "miss"	True positive (TP)	P
		N*	p*	

Precision:

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

# Confusion Matrices

		Predicted class		
		Negative	Positive	
True class	Negative	True negative (TN)	False positive (FP) “false alarm”	N
	Positive	False negative (FN) “miss”	True positive (TP)	P
		N*	p*	

Recall  
(True Positive Rate):

$$TP/(FN+TP) = TP/P$$

# Confusion Matrices

		Predicted class		
		Negative	Positive	
True class	Negative	True negative (TN)	False positive (FP) "false alarm"	N
	Positive	False negative (FN) "miss"	True positive (TP)	P
		N*	p*	

False Positive Rate:

$$FP/(TN+FP) = FP/N$$

# Precision and Recall

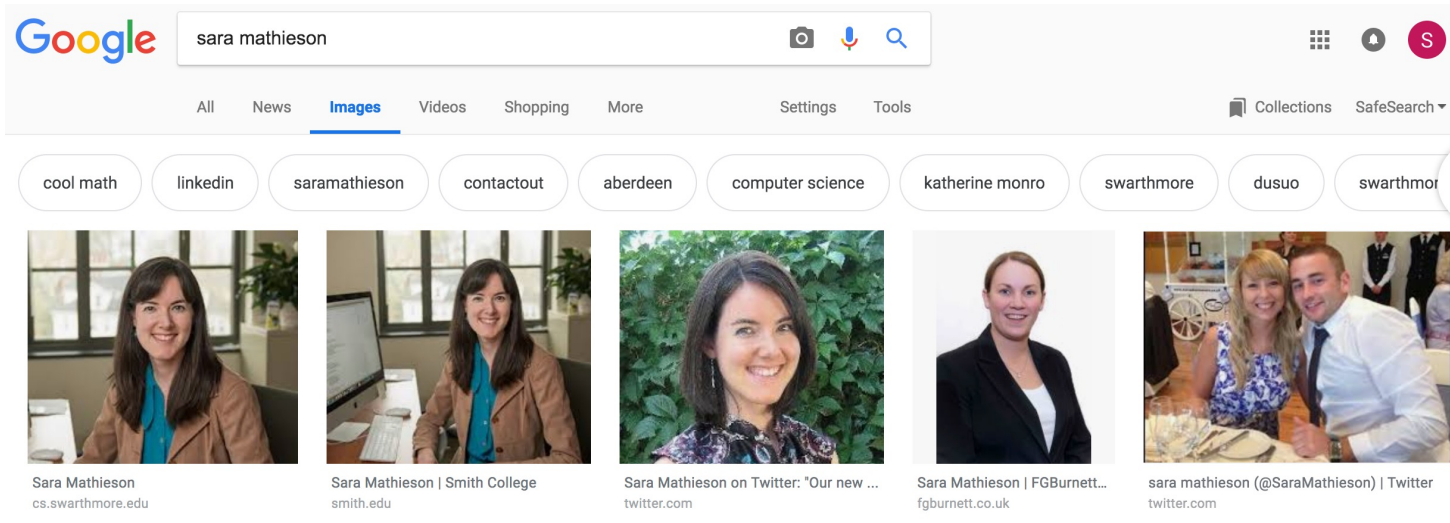
- Precision: of all the “flagged” examples, which ones are actually relevant (i.e. positive)?

(Purity)

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

(Completeness)

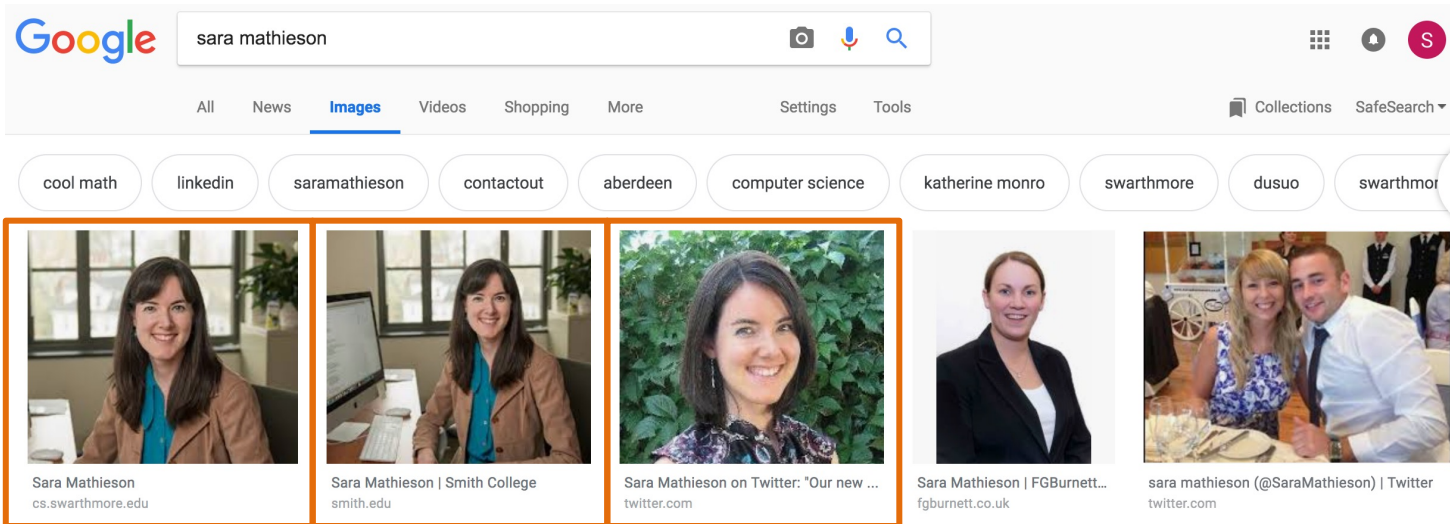
# Precision and Recall



$P=6$  (number of images that are actually me)

- Precision?
- Recall?

# Precision and Recall

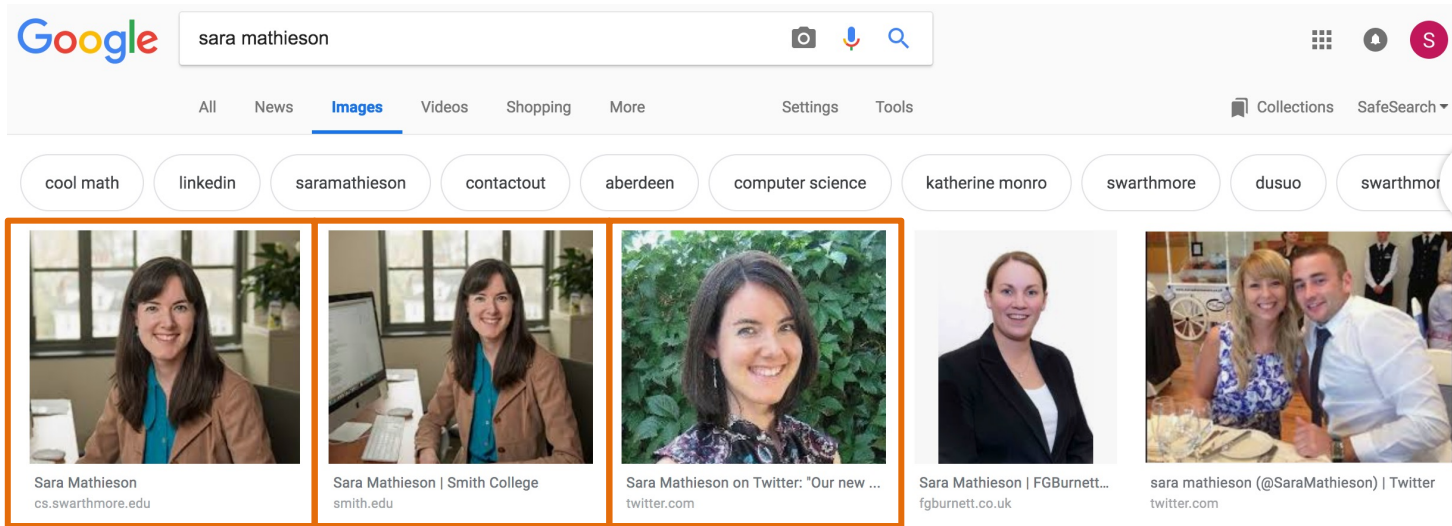


$P=6$  (number of images that are actually me)

- Precision =  $TP/(FP+TP) = 3/5$
- Recall?



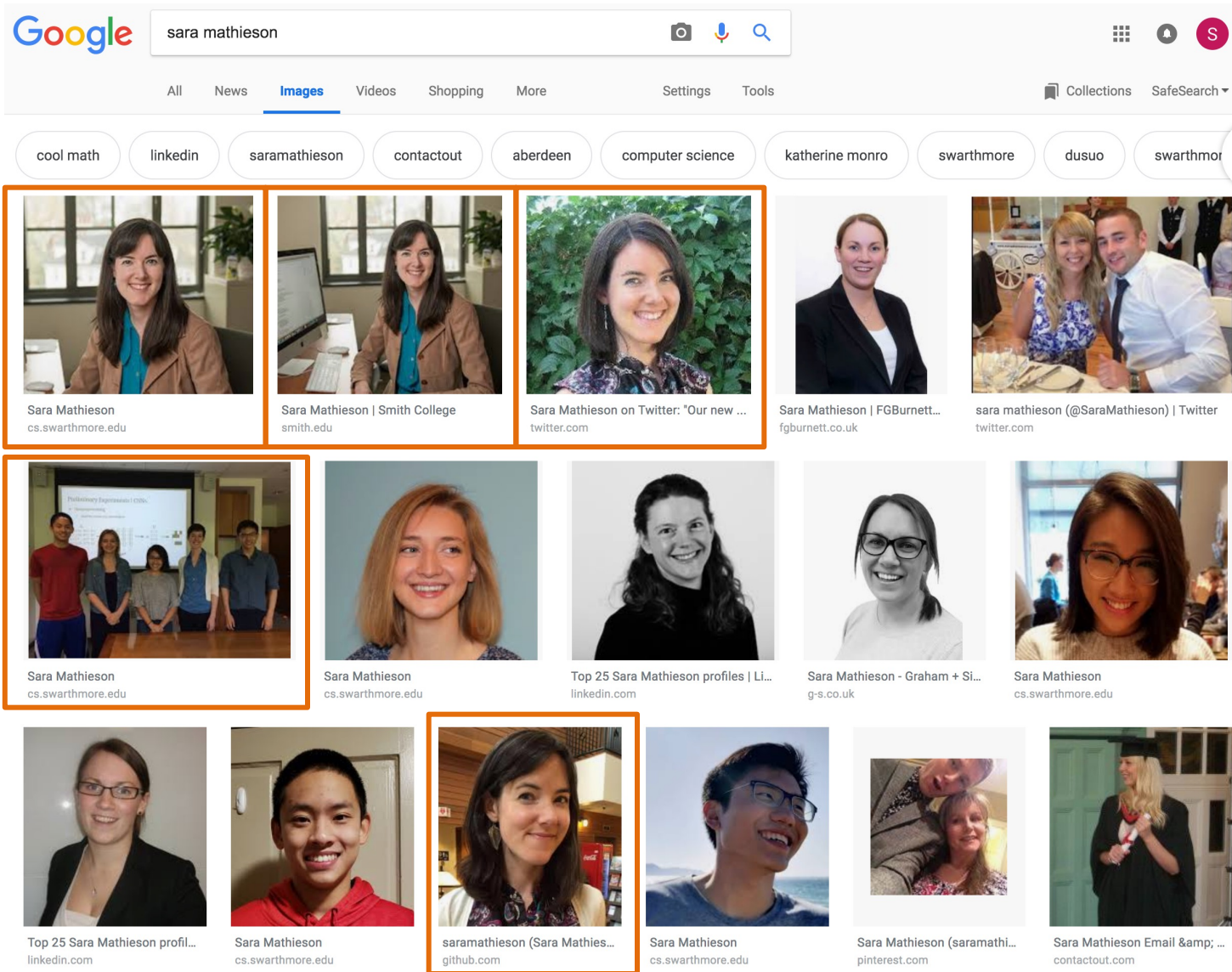
# Precision and Recall



$P=6$  (number of images that are actually me)

- Precision =  $TP/(FP+TP) = 3/5$
- Recall =  $TP/(FN+TP) = 3/6$

# Precision and Recall

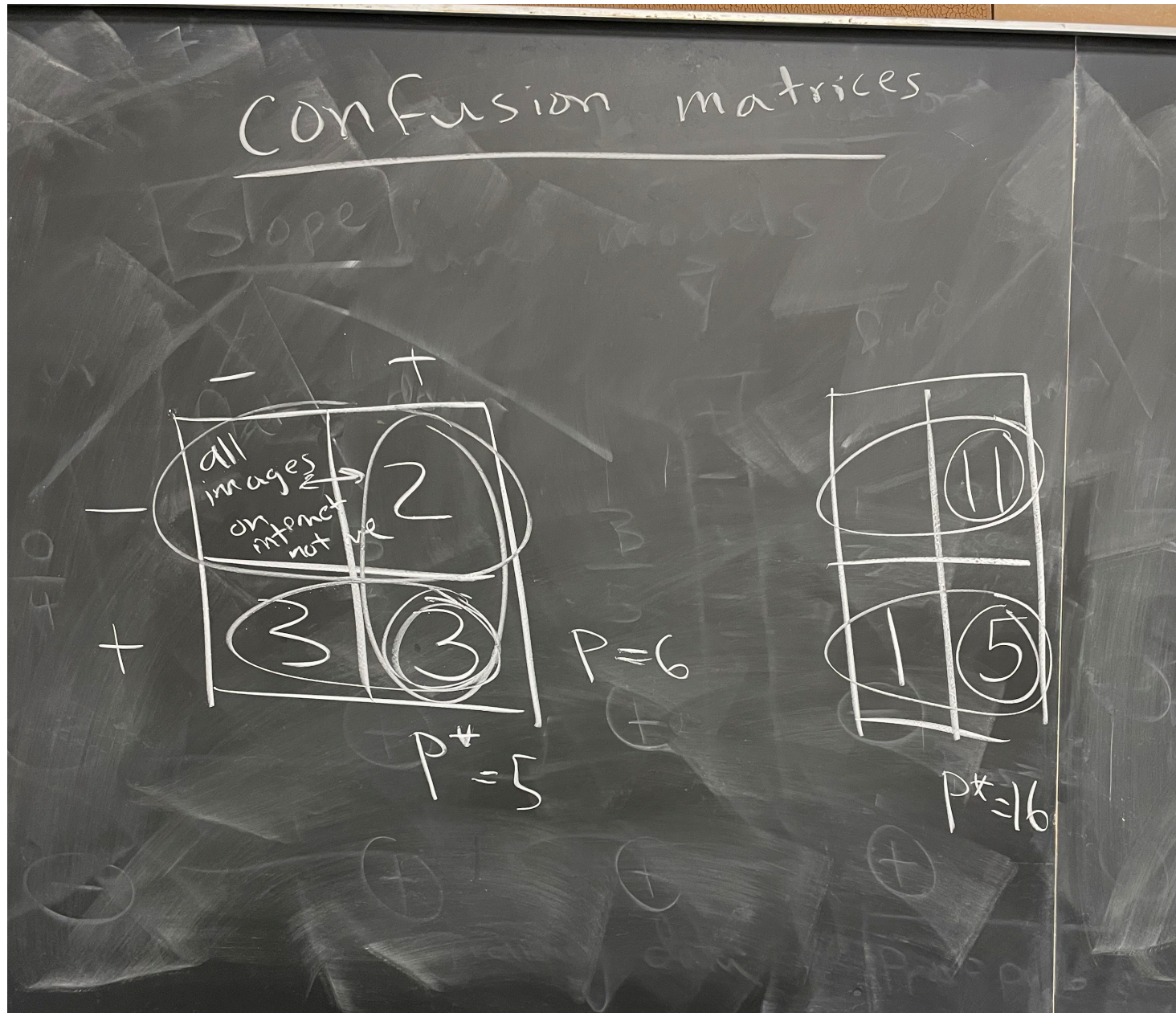


$P=6$  (number of images that are actually me)

- Precision =  $5/16$
- Recall =  $5/6$

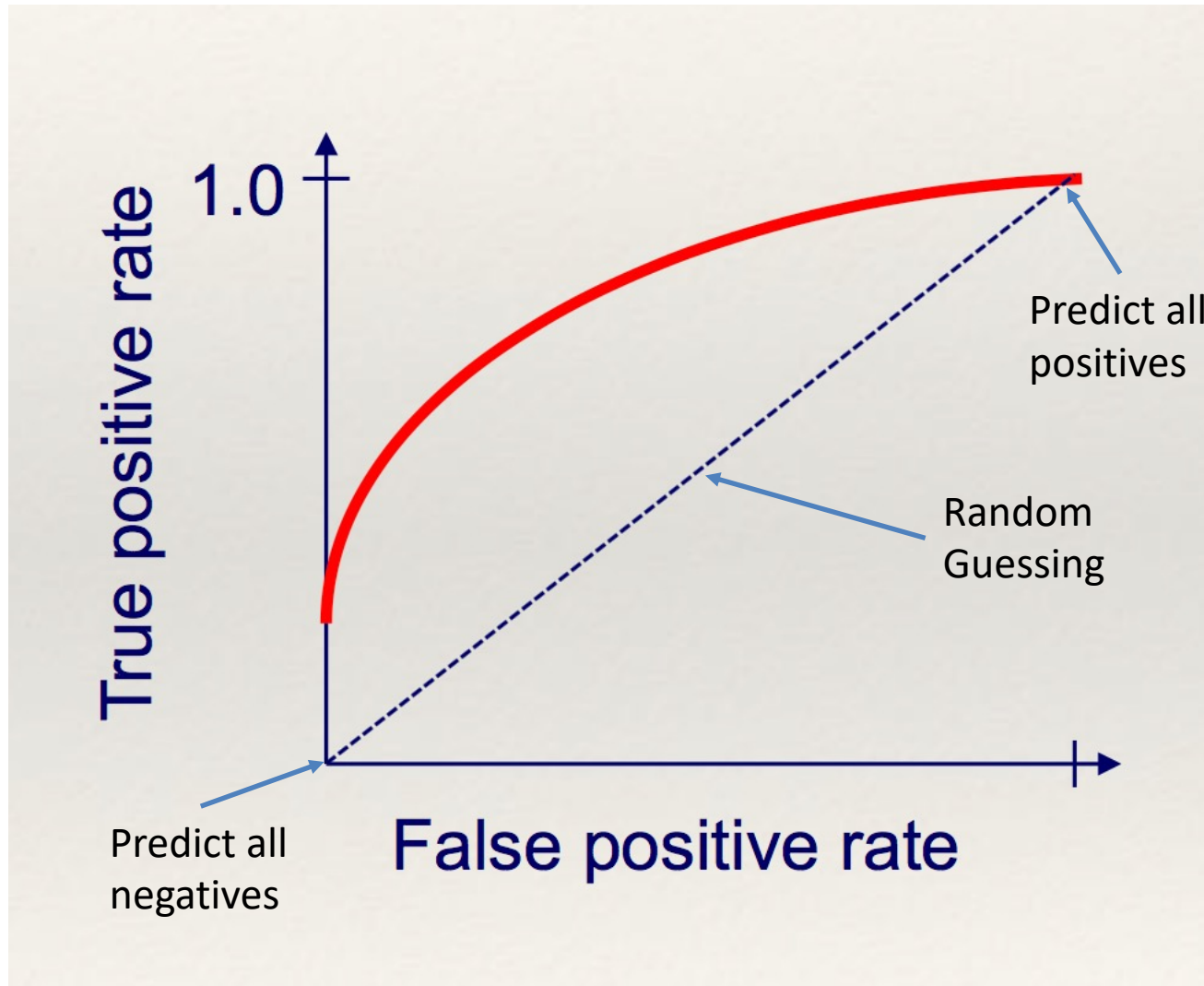


# Confusion matrices for google search example

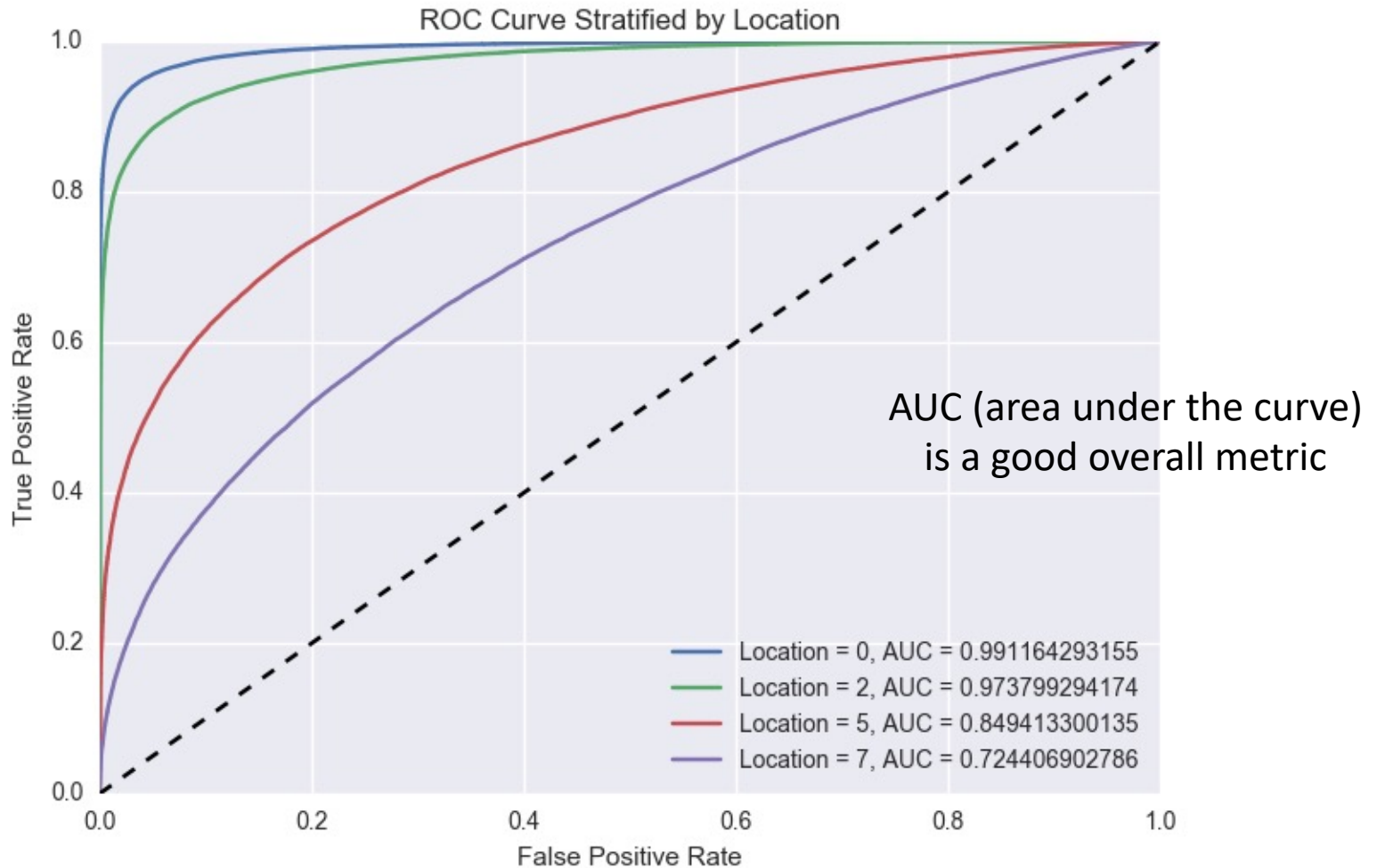


# ROC curve (Receiver Operating Characteristic)

More history here! [https://en.wikipedia.org/wiki/Receiver\\_operating\\_characteristic](https://en.wikipedia.org/wiki/Receiver_operating_characteristic)



# ROC curve example: comparing methods



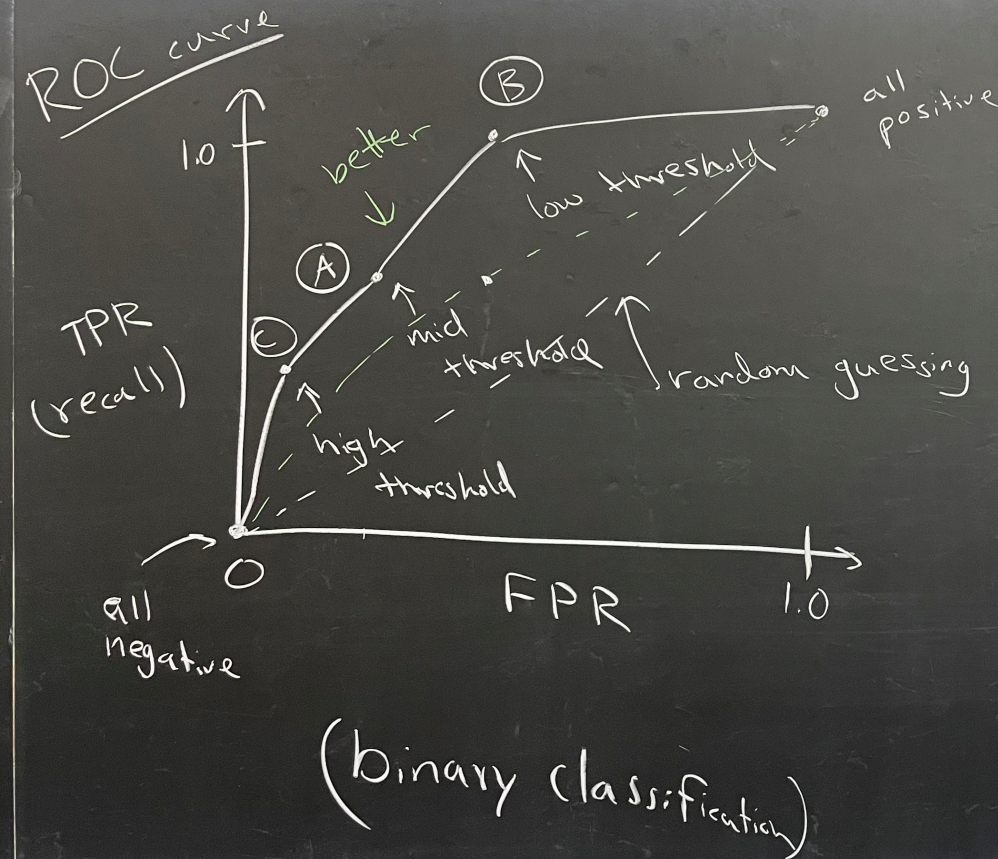
Example of a ROC curve from my research  
Chan, Perrone, Spence, Jenkins, Mathieson, Song



# How to get a ROC curve for probabilistic methods?

- Usually we use 0.5 as a threshold for binary classification
- Vary the threshold! (i.e. choose 0, 0.1, 0.2,...)
  - $P(y=1 \mid x) \geq 0.2$   $\Rightarrow$  classify as 1 (positive)
  - $P(y=1 \mid x) < 0.2$   $\Rightarrow$  classify as 0 (negative)

# ROC curve example



$$\textcircled{A} \text{ FPR} = \frac{15}{65+15} = \frac{15}{80}$$

$$\text{TPR} = \frac{13}{7+13} = \frac{13}{20}$$

$$\textcircled{B} \text{ FPR} = \frac{30}{80}$$

$$\text{TPR} = \frac{19}{20}$$

$$\textcircled{C} \text{ FPR} = \frac{4}{80}$$

$$\text{TPR} = \frac{9}{20}$$

# Quote of the week

“If you want to go fast, go alone. If you want to go far, go together.” --African proverb