CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2019

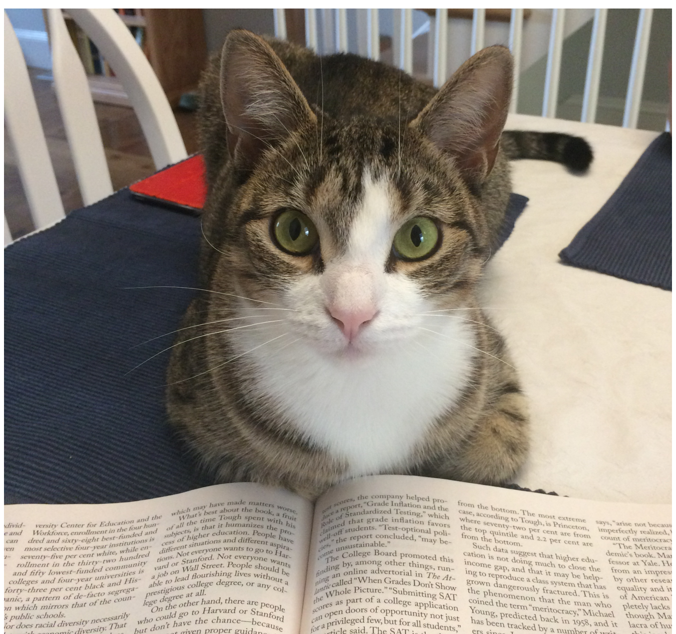
HAVERFORD
COLLEGE

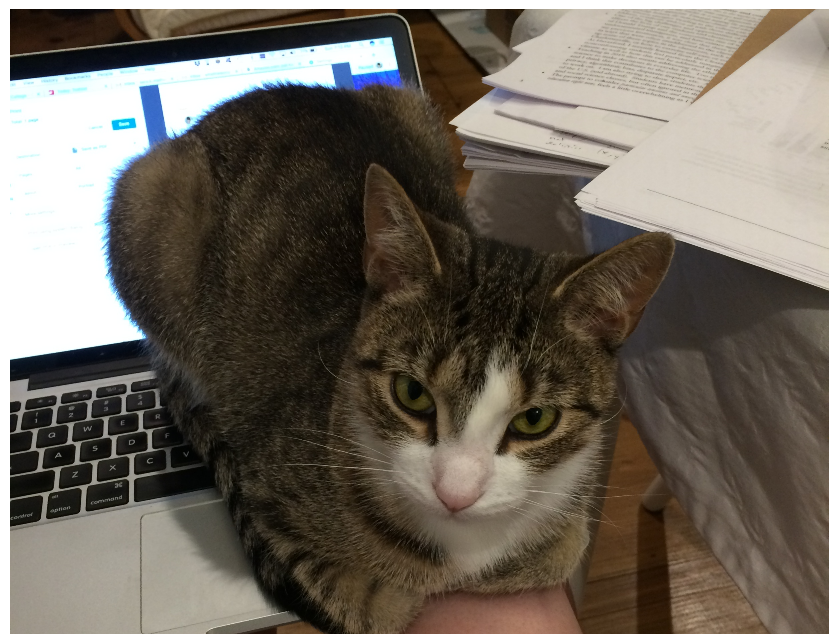
Admin

- **Lab 5 TODAY!**
  - Office hours today 12:30—1:30pm (H110)
- **Reading Quiz Thursday** (Duame Section 13.1)
- Lab 6 due Friday Nov 1
  - Checkpoint during lab on Thursday Oct 31 (Part 1 and 2)

In lab Thursday

- Hand back the midterm
- Go over common issues
- Start Lab 6





Outline for October 22

- Evaluation metrics
  - Confusion matrices revisited
  - ROC curves
  - Relationship to probabilistic methods
- Ensemble methods
  - Bagging
  - Random forests

Outline for October 22

- Evaluation metrics
  - Confusion matrices revisited
  - ROC curves
  - Relationship to probabilistic methods
- Ensemble methods
  - Bagging
  - Random forests

For now: assume binary classification task

- Transactions that indicate credit card fraud
- 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”)

For now: assume binary classification task

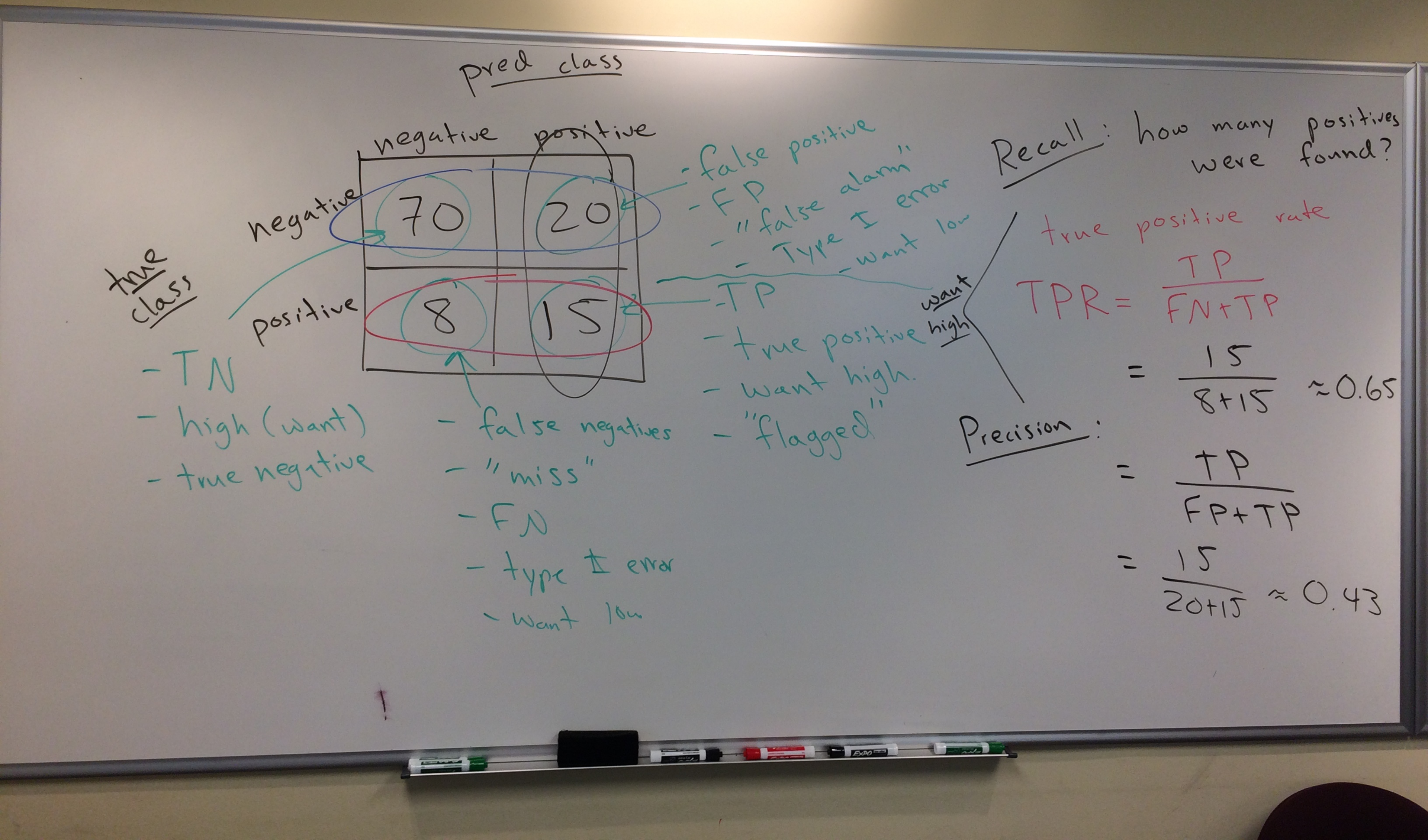
- Transactions that indicate credit card fraud
- Detecting which scans show tumors
- Prenatal test for Down’s Syndrome
- Finding genes under natural selection
- Finding regions of the genome with high recombination rate (“hotspots”)

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

Goals of Evaluation

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

Back to Confusion Matrices…



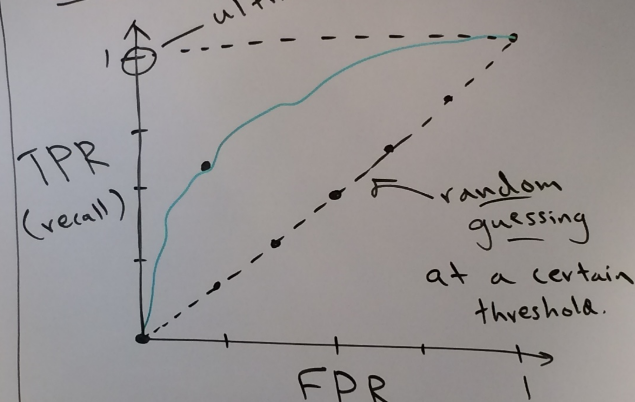
false positive rate

$$FPR = \frac{FP}{FP + TN}$$

want <u>low</u>

$$= \frac{20}{70+20} \approx 0.22$$

ROC curve



0	90
0	23

90 = N

23 = P

predict all positive

$N^* = 0 \quad P^* = 113$

$$FPR = \frac{90}{0+90} = 1$$

$$TPR = \frac{23}{0+23} = 1$$

90	0
23	0

FPR = 0

TPR = 0

predict all negative

45	45
12	11

$$TPR = \frac{11}{23} \approx 0.5$$

$$FPR = \frac{45}{90} = 0.5$$

Probalistic Model

only at test time

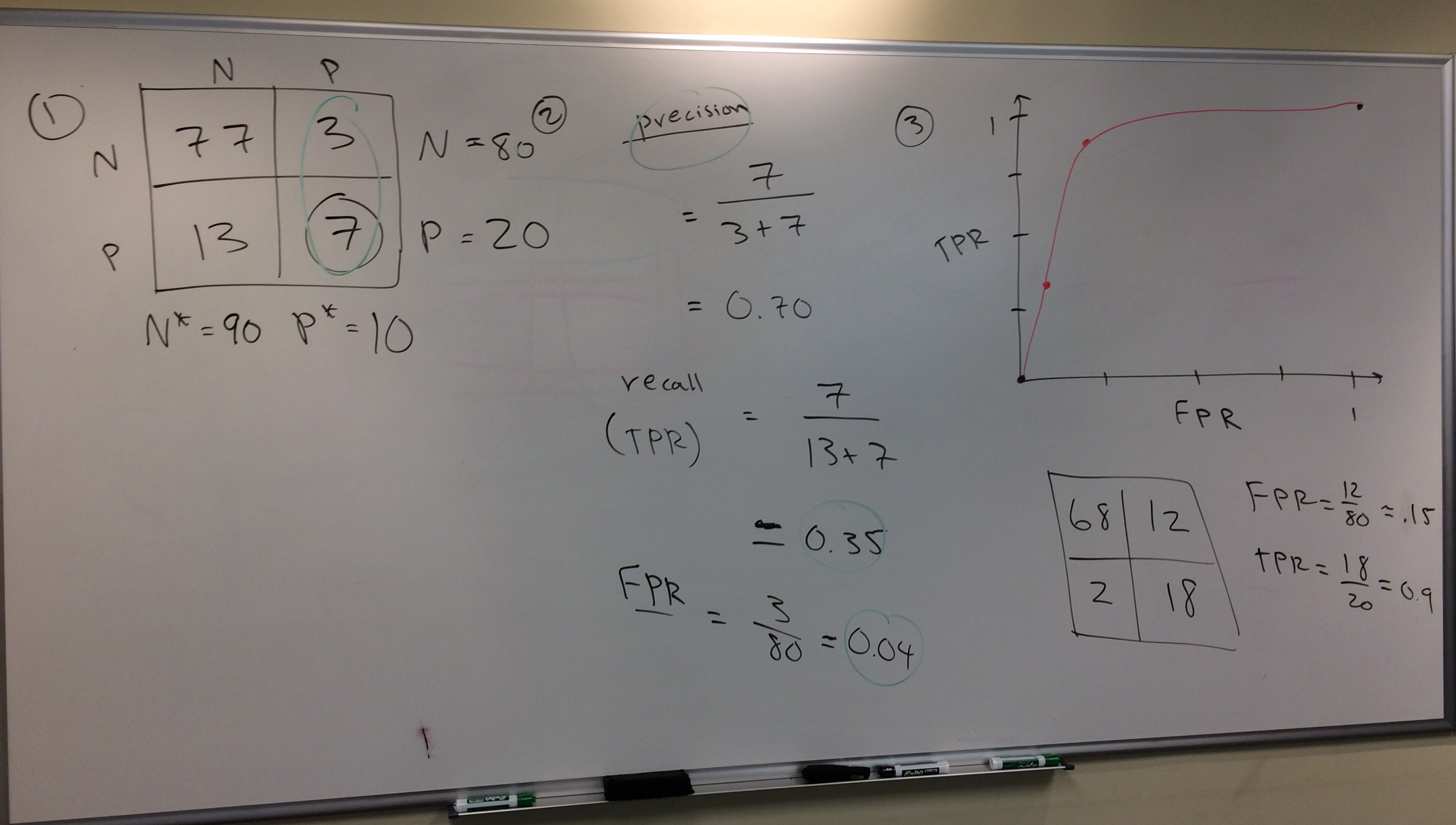
threshold: 0.25

$$p(y=1|x) \begin{cases} > 0.25 \Rightarrow \hat{y}=1 \\ \leq 0.25 \Rightarrow \hat{y}=0 \end{cases}$$

threshold: 0.75

want to be confident

Handout 12



①
N P
N 77 3
P 13 7
N = 80
P = 20
N* = 90 P* = 10
②
precision
= 7 / (3+7)
= 0.70
recall (TPR) = 7 / (13+7)
= 0.35
FPR = 3/80 = 0.04
③
1
TPR
FPR
1
68 12
2 18
FPR = 12/80 ≈ .15
TPR = 18/20 = 0.9

Precision and Recall

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

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

Modified from Ameet Soni

Precision and Recall

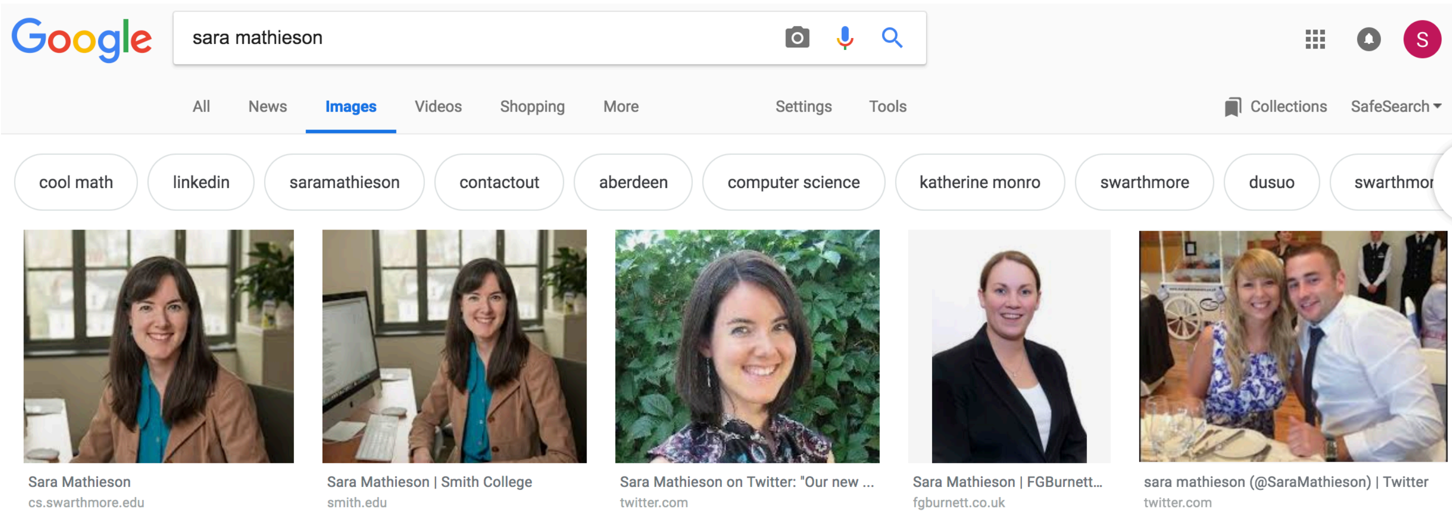
- Precision: of all the "flagged" examples, which ones are actually relevant (i.e. positive)?

(Purity)

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

(Completeness)

Precision and Recall

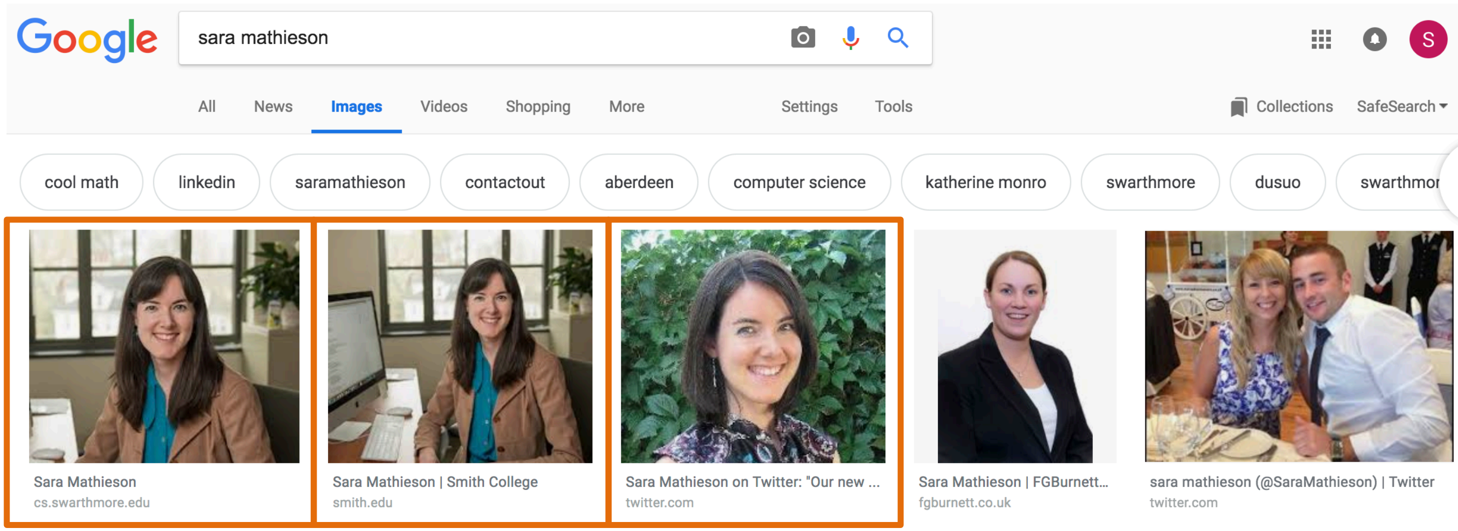


P=6 (number of images that are actually me)

- Precision?
- Recall?

Modified from Ameet Soni

Precision and Recall

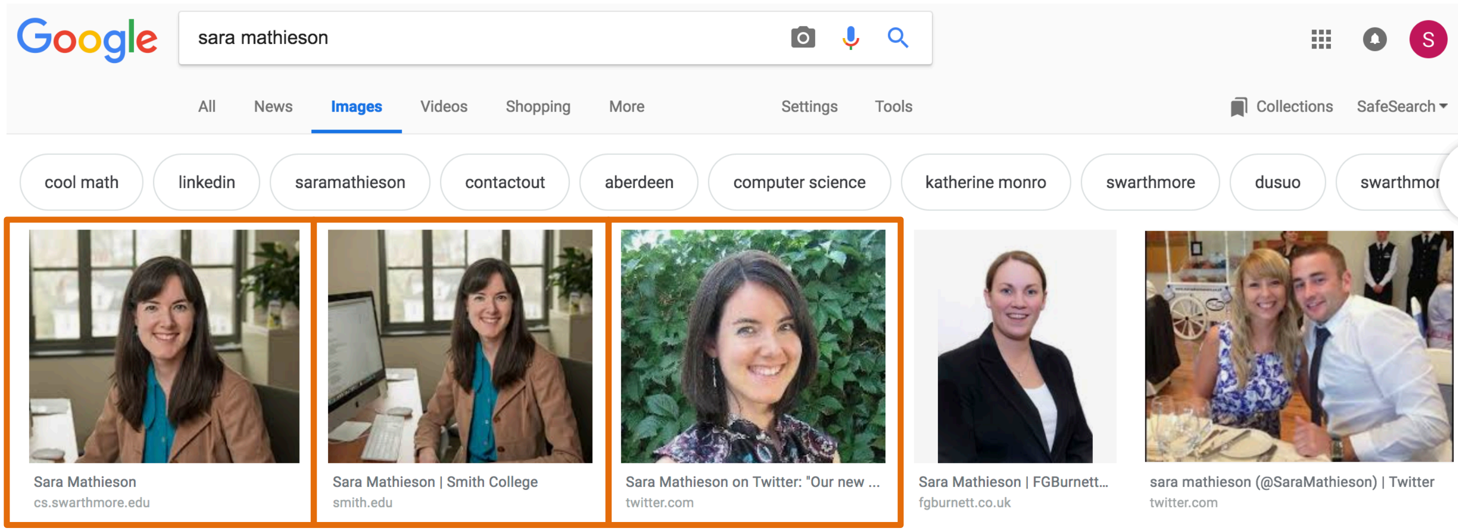


P=6 (number of images that are actually me)

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

Modified from Ameet Soni

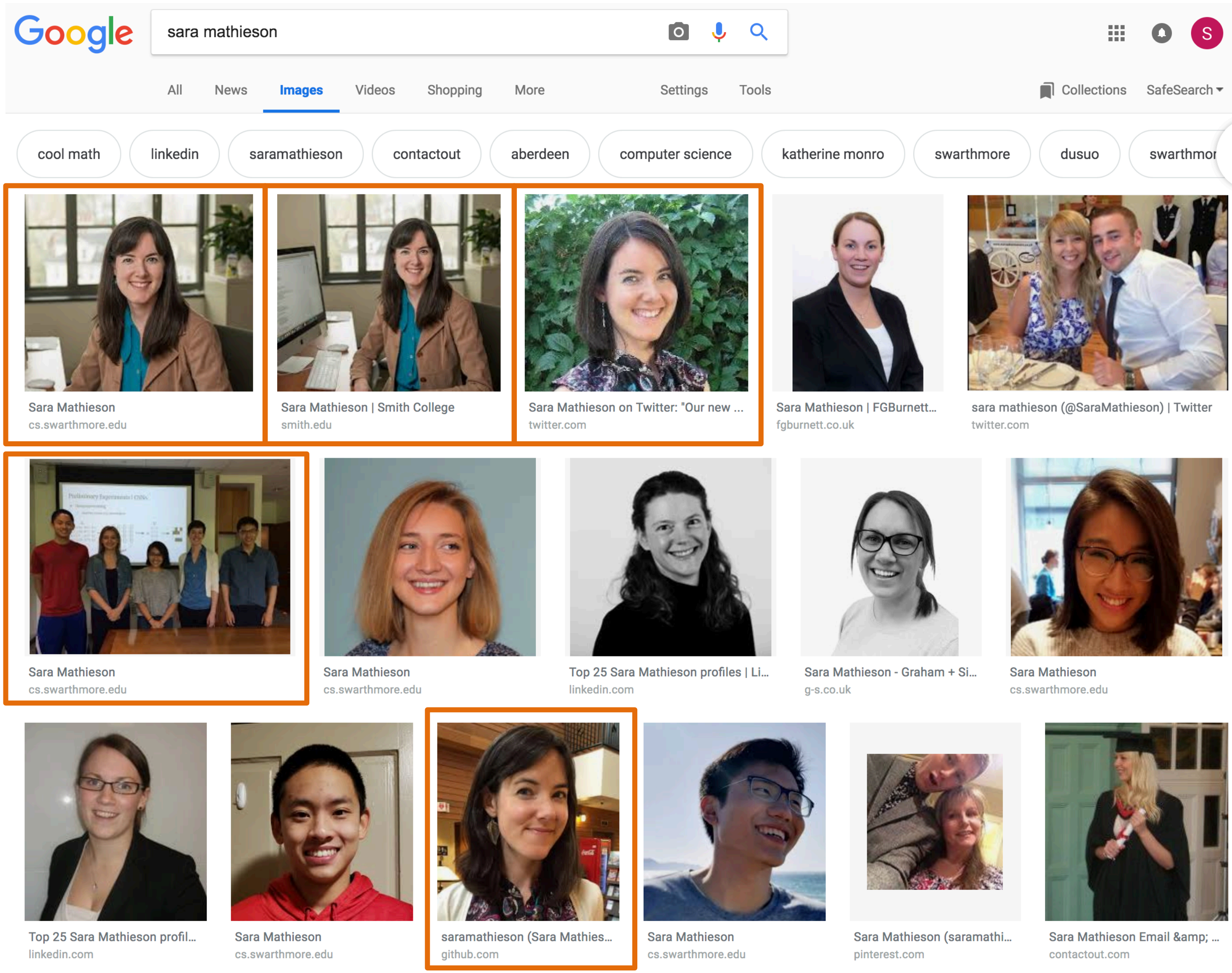
Precision and Recall



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

Modified from Ameet Soni

Recap Confusion Matrices

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

Modified from Jessica Wu

Recap Confusion Matrices

Predicted class

True class

	Negative	Positive	
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”)	

Modified from Jessica Wu

Recap 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*	

Modified from Jessica Wu

Recap 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*	

Error:

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

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

Modified from Jessica Wu

Recap Confusion Matrices

Predicted class

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

Accuracy = 1-Error:

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

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

Modified from Jessica Wu

Recap 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*

Modified from Jessica Wu

Recap Confusion Matrices

True class	Predicted class: Negative	Predicted class: Positive	
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

Recap Confusion Matrices

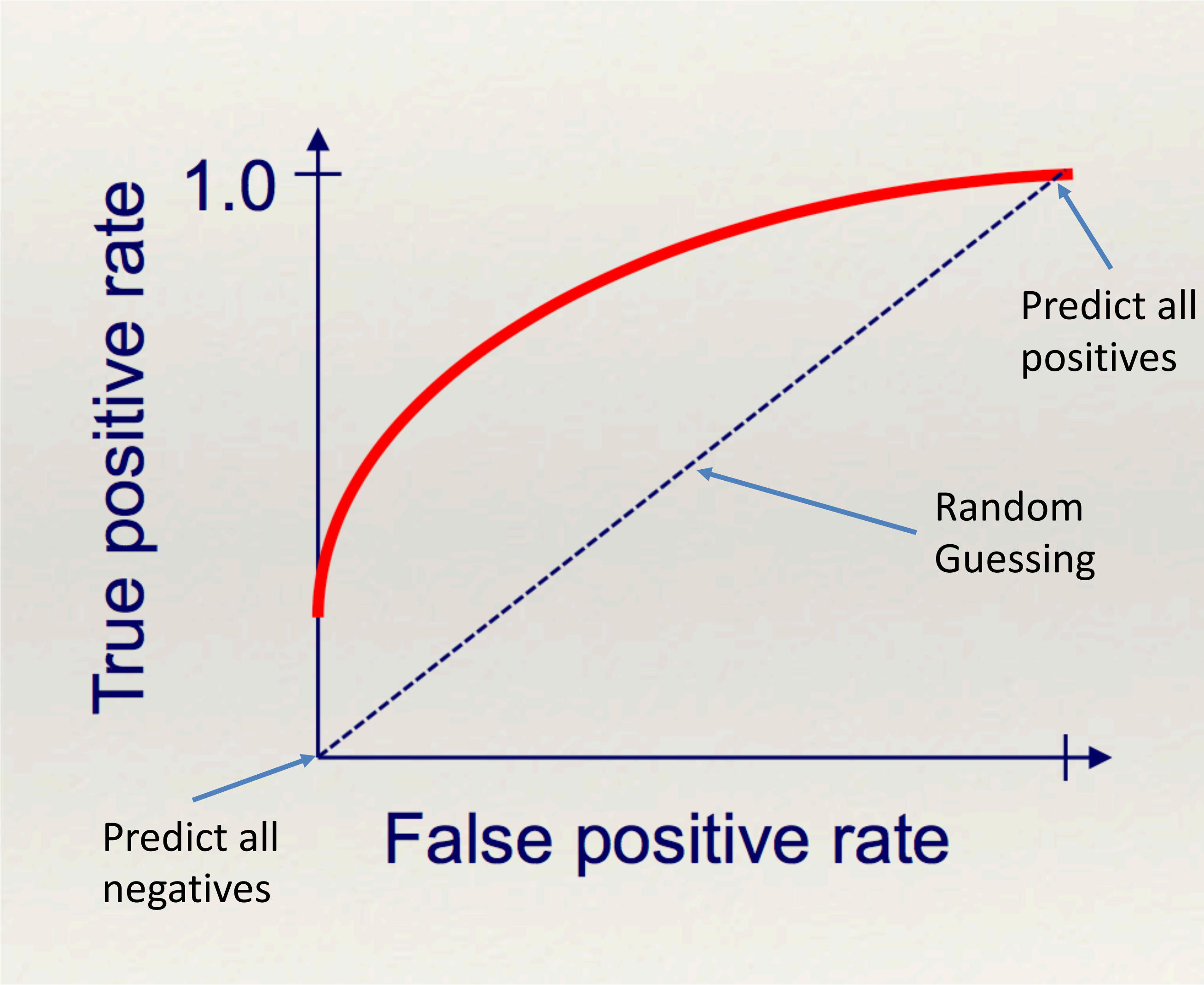
True class	Predicted class: Negative	Predicted class: Positive	
Negative	True negative (TN)	False positive (FP) “false alarm”	N
Positive	False negative (FN) “miss”	True positive (TP)	P
	N*	P*	

False Positive Rate:

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

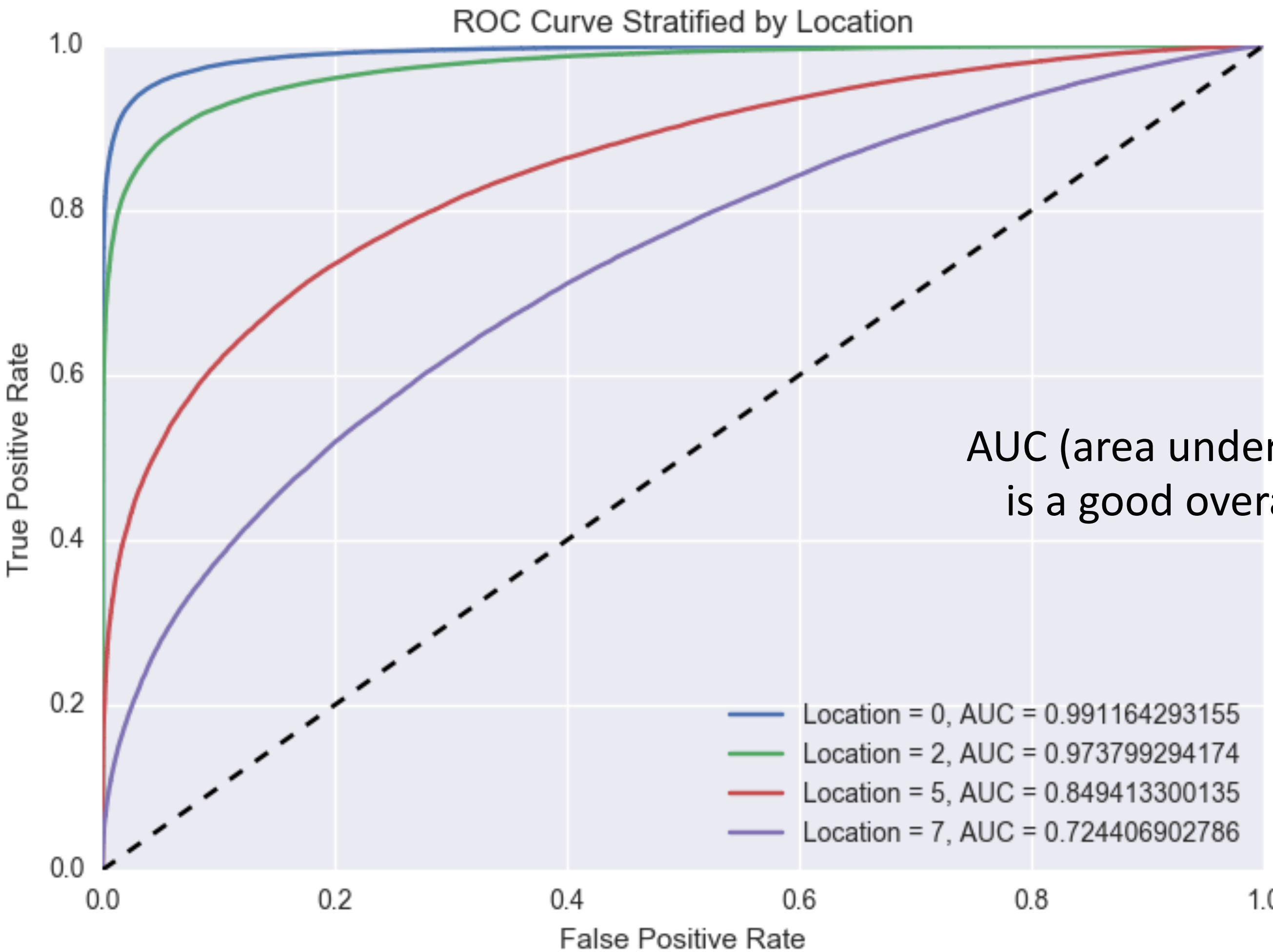
Modified from Jessica Wu

ROC curve (Receiver Operating Characteristic)



Modified from Ameet Soni

ROC curve example: comparing methods



AUC (area under the curve)
is a good overall metric

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 0.25)

  – $P(y=1 \mid x) > 0.25$ => classify as 1 (positive)

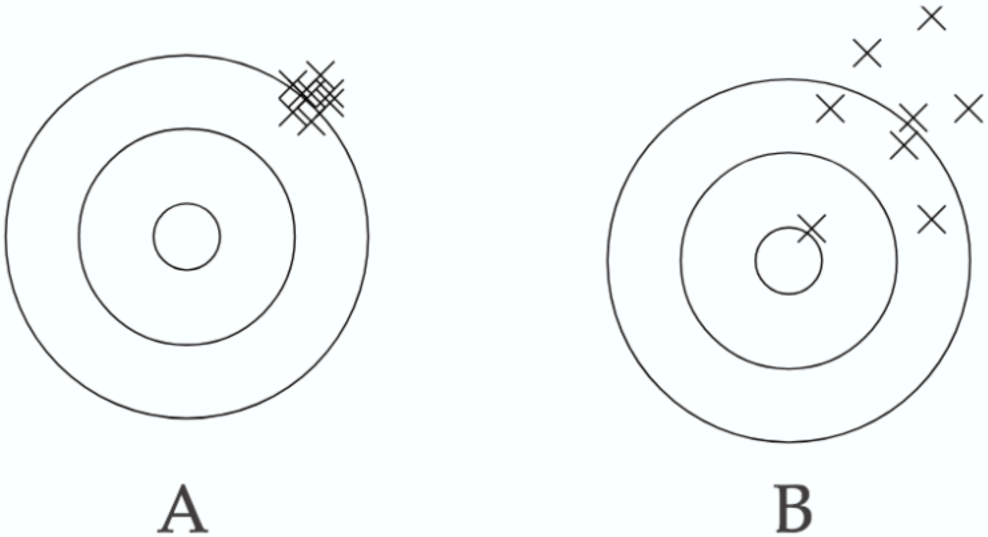
  – $P(y=1 \mid x) <= 0.25$ => classify as 0 (negative)

Outline for October 22

- Evaluation metrics
  - Confusion matrices revisited
  - ROC curves
  - Relationship to probabilistic methods

- Ensemble methods
  - Bagging
  - Random forests

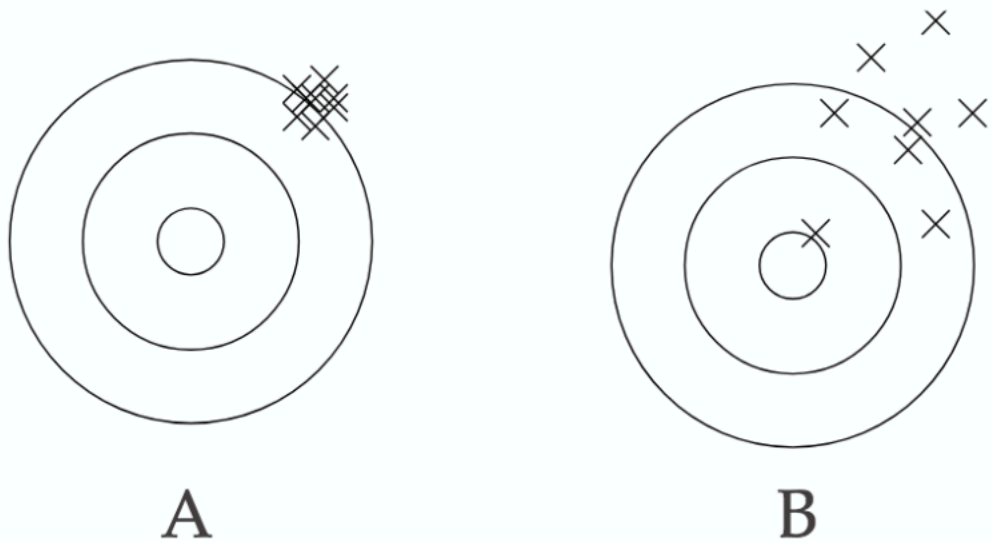
Quiz: recap bias and variance



Label each picture with variance (high or low) and bias (high or low)

Example from Ameet Soni

Quiz: recap bias and variance

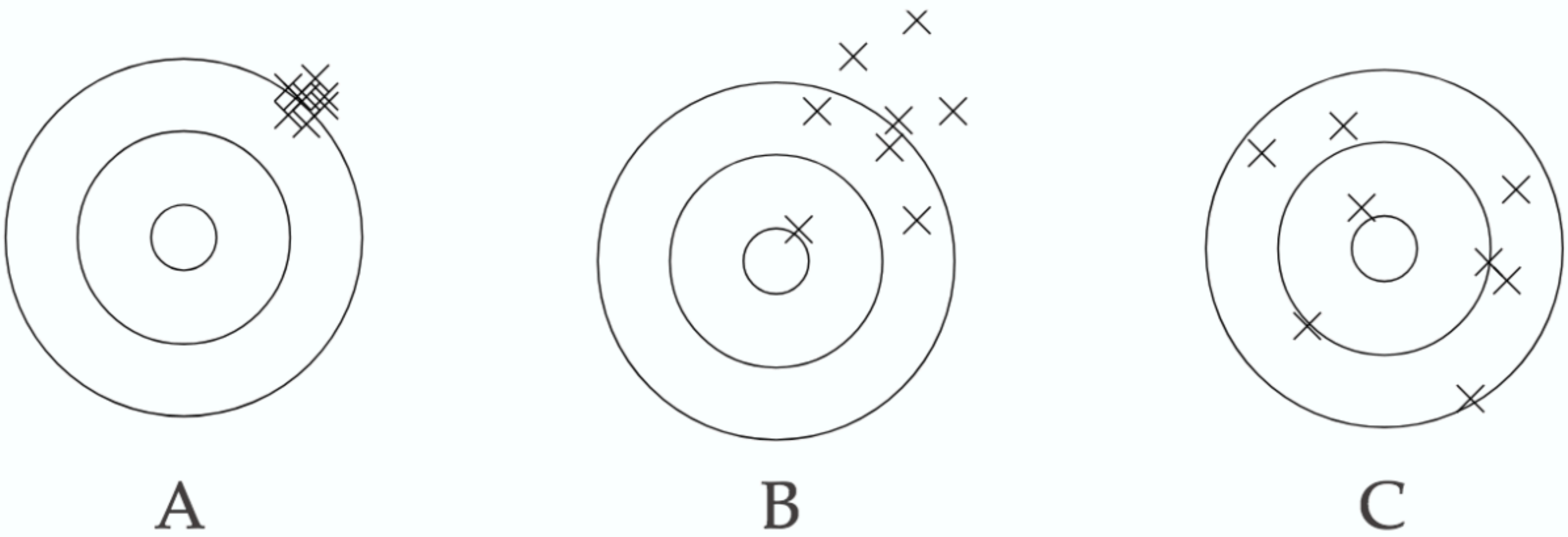


Variance: low
Bias: high

Label each picture with variance (high or low) and bias (high or low)

Example from Ameet Soni

Quiz: recap bias and variance



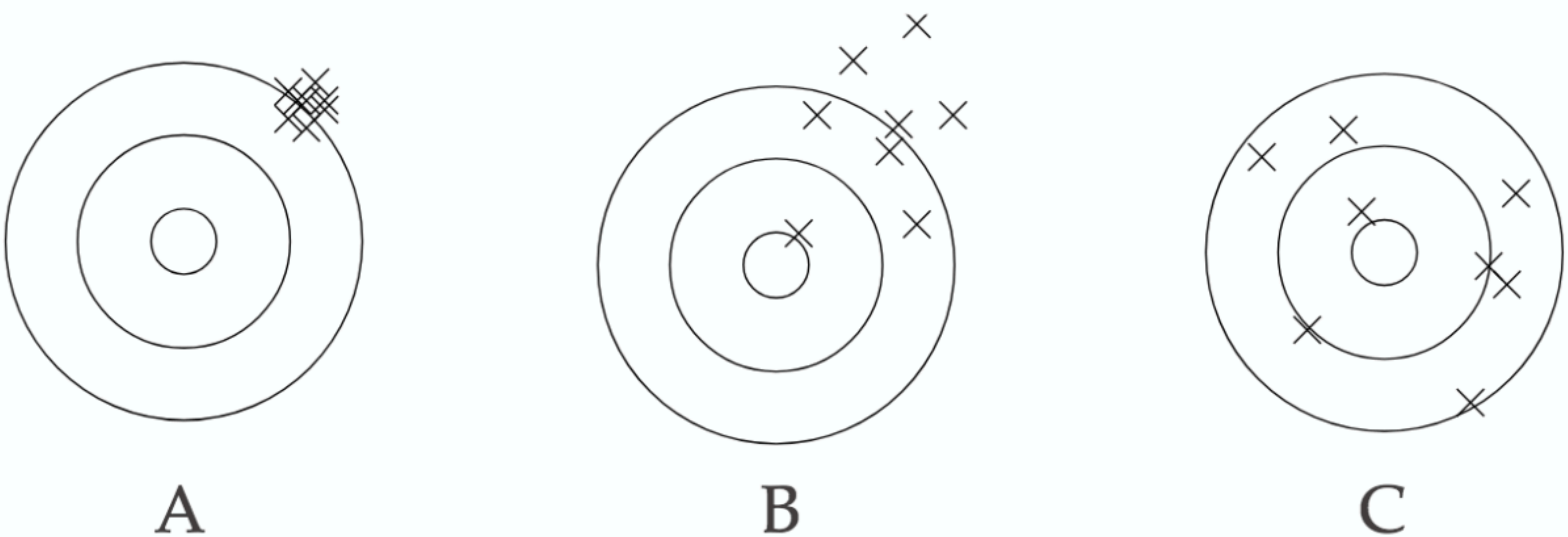
Variance: low
Bias: high

Variance: high
Bias: high

Label each picture with variance (high or low) and bias (high or low)

Example from Ameet Soni

Quiz: recap bias and variance

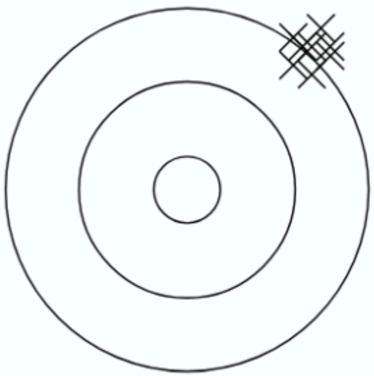


Variance:	low	Variance:	high	Variance:	high
Bias:	high	Bias:	high	Bias:	low

Label each picture with variance (high or low) and bias (high or low)

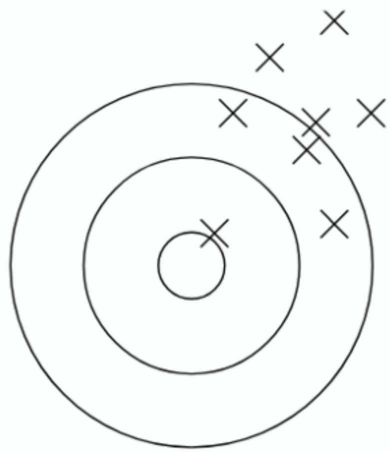
Example from Ameet Soni

Quiz: recap bias and variance



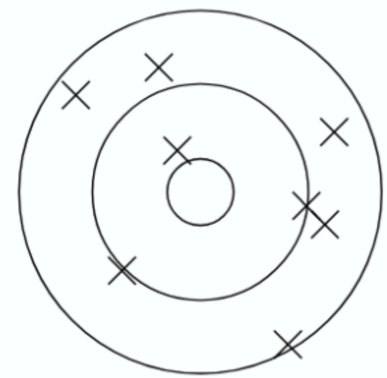
A

Variance: low
Bias: high



B

Variance: high
Bias: high



C

Variance: high
Bias: low

This is the type of classifier we want to average!

Label each picture with variance (high or low) and bias (high or low)

Example from Ameet Soni

Ensemble Idea

- Average the results from several models with high variance and low bias
  - Important that models be diverse (don't want them to be wrong in the same ways)

- If *n* observations each have variance s^2, then the mean of the observations has variance s^2/n (reduce variance by averaging!)

Learning Theory

Let *H* be the hypothesis space

Three sources of limitations for traditional classifiers:

- <u>Statistical</u> - *H* is too large relative to size of data
  - Many hypotheses can fit the data by chance
- <u>Computational</u> - *H* is too large to completely search for "best" model
- <u>Representational</u> - *H* is not expressive enough

Learning Theory

- Statistical: Average of unstable models (high variance) has more stability
- Computational: searching from multiple starting points is better approximation than one starting point
- Representational: sum of many models can represent more hypotheses than an individual model

Modified from Ameet Soni

Learning Theory

- Statistical: Average of unstable models (high variance) has more stability
- Computational: searching from multiple starting points is better approximation than one starting point
- Representational: sum of many models can represent more hypotheses than an individual model

Ensembles can address all 3!

Modified from Ameet Soni

Learning Theory

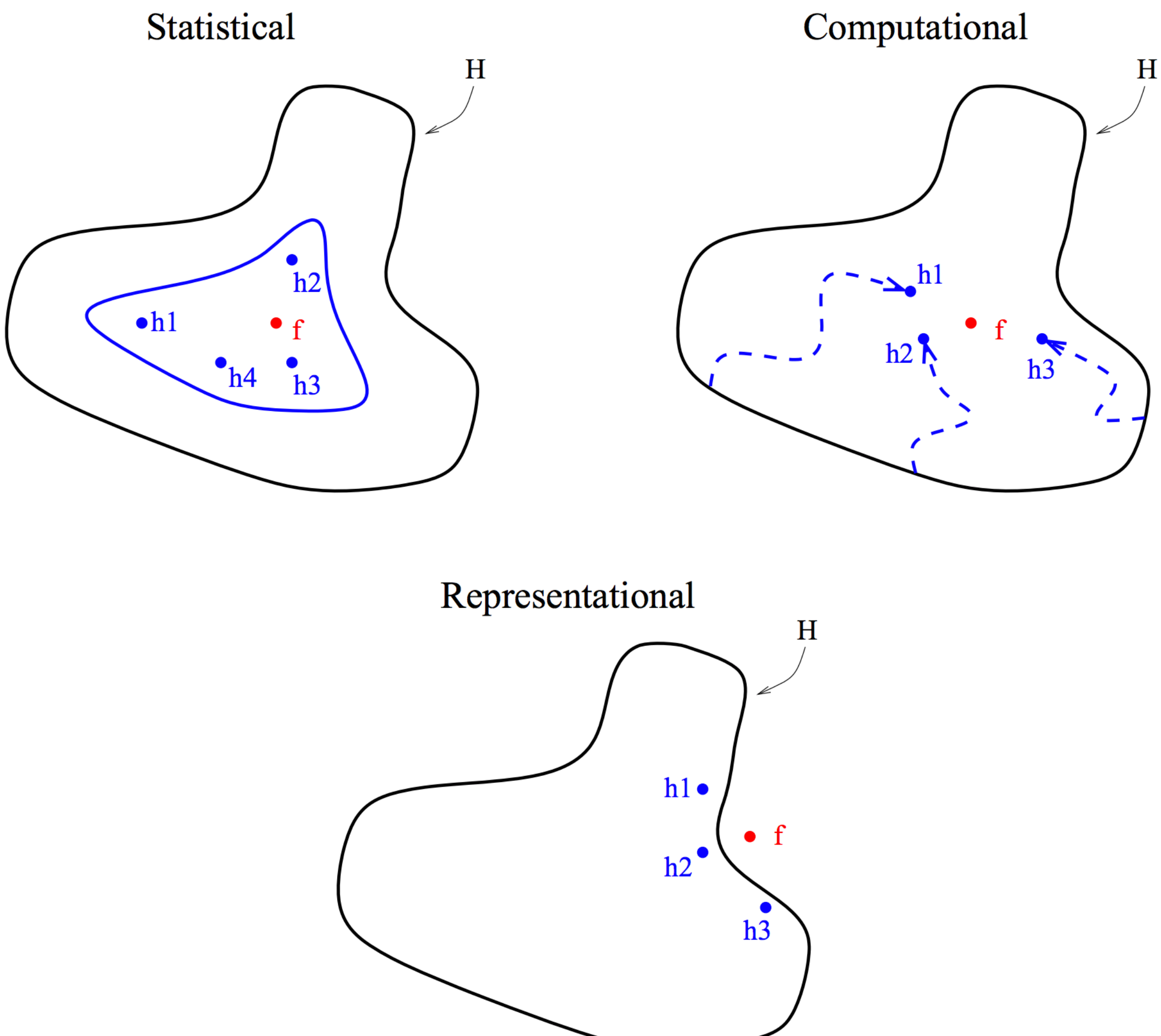
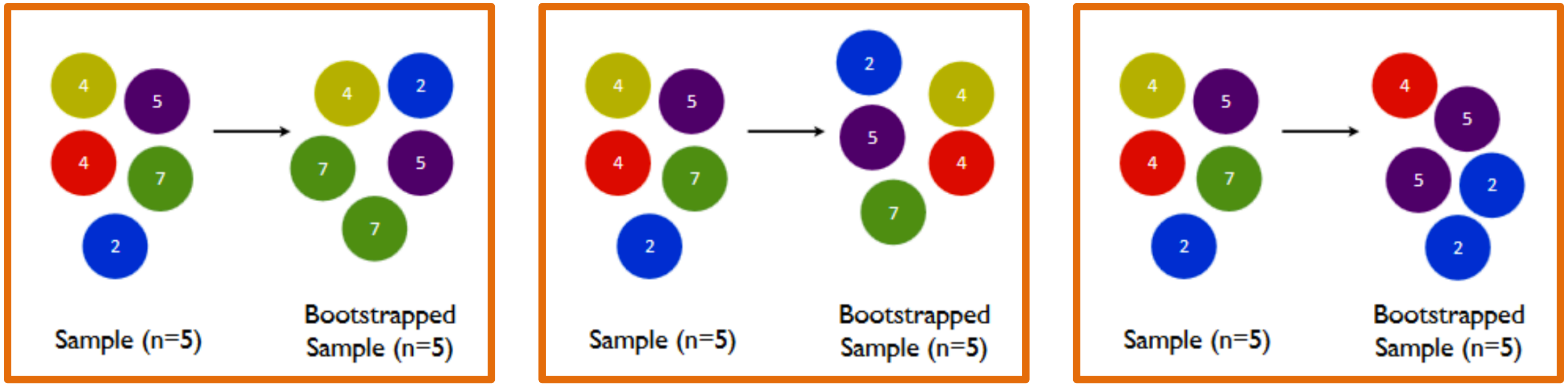


Figure from Tom Dietterich

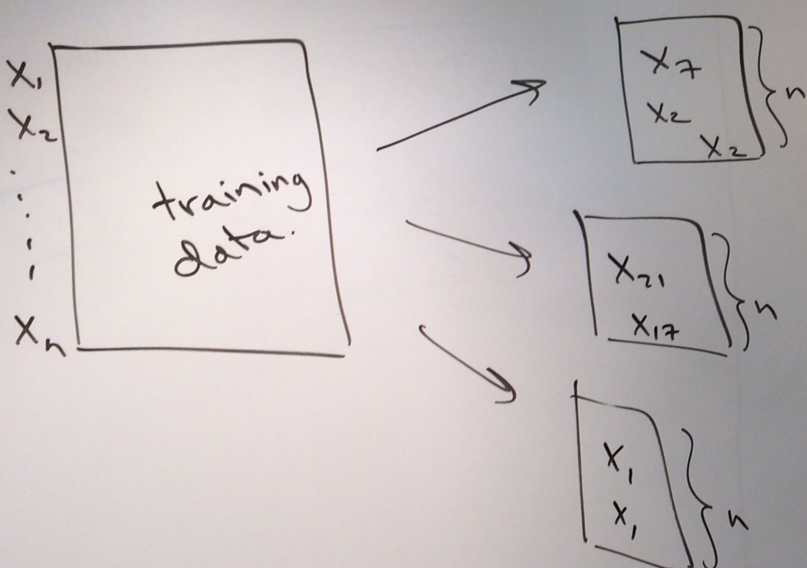
Bagging Algorithm

- Bagging = Bootstrap Aggregation [Brieman, 1996]
- *Bootstrap* (randomly sample with replacement) original data to create many different training sets
- Run base learning algorithm on each new data set independently



Desmond Ong, Stanford

Bootstrap   sample with replacement



prob we don't choose an example

$$\left(\frac{n-1}{n}\right)^n = \left(1-\frac{1}{n}\right)^n \xrightarrow[\substack{\text{lim}\\ \text{as}\\ n\to\infty}]{} e^{-1} \approx \underset{\text{large}}{\underline{0.38}}$$

$$\lim_{n\to\infty}\left(1+\frac{x}{n}\right)^n = e^x$$

Notation

For Ensembles

T = # models/classifiers (index t)

x = test example (could be vector)

$X^{(t)}$ = bootstrap training set t

$h^{(t)}(x)$ = hypothesis about x from model t

r = prob. of error of individual model

R = # votes for wrong class.

Bagging (Bootstrap Aggregation) , $y \in \{0,1\}$

Train for t in range(T):

- create bootstrap dataset $X^{(t)}_{(n \times p)}$
- train on $X^{(t)}$ to get model $h^{(t)}$

also in $\{0,1\}$

threshold already applied

Test for x in test data:

$$h(x) = \underset{y \in \{0,1\}}{\operatorname{argmax}} \sum_{t=1}^{T} \mathbb{1}\left(h^{(t)}(x) = y\right)$$