CMSC 471: Machine Learning

Evaluation

Many slides courtesy Frank Ferraro

Evaluation methodology (1)

Standard methodology:

1. Collect large set of examples with correct classifications (aka ground truth data)
2. Randomly divide collection into two disjoint sets: ***training*** and ***test*** *(e.g., via a 90-10% split)*
3. Apply learning algorithm to **training** set giving hypothesis
4. Measure performance of H on the held-out **test** set

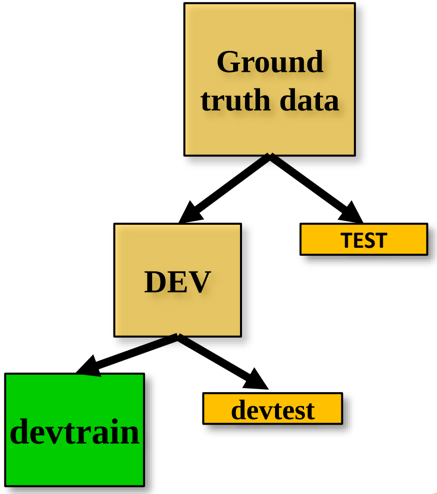
Evaluation methodology (2)

- Important: keep the training and test sets **disjoint**!
- Study efficiency & robustness of algorithm: repeat steps 2-4 for **different training sets & training set sizes**
- On modifying algorithm, **restart with step 1** to avoid evolving algorithm to work well on just this collection

Evaluation methodology (3)

Common variation on methodology:

1. Collect set of examples with correct classifications
2. Randomly divide it into two disjoint sets: ***development*** & ***test***; further divide development into ***devtrain*** & ***devtest***
3. Apply ML to *devtrain*, giving hypothesis H
4. Measure performance of H w.r.t. *devtest* data
5. Modify approach, repeat 3-4 as needed
6. Final test on *test* data

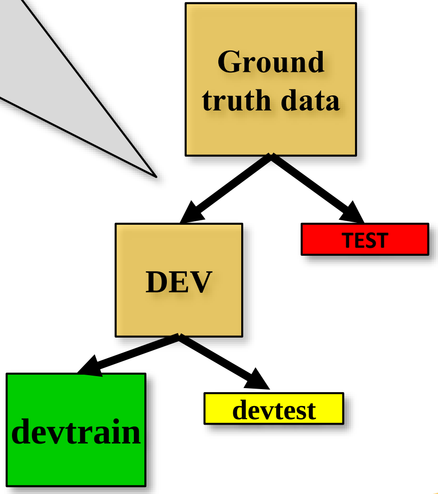


Evaluation methodology (4)

- Only **devtest** data used for evalua-tion during system **development**
- When all development has ended, **test** data used for **final evaluation**
- Ensures final system not influenced by test data
- If more development needed, get new dataset!

Comm

1. Colle ... ifications
2. Rand ... *deve* ... opment into
3. Appl
4. Mea ... *devtest* data
5. Modify approach, repeat 3-4 as needed
6. Final test on *test* data

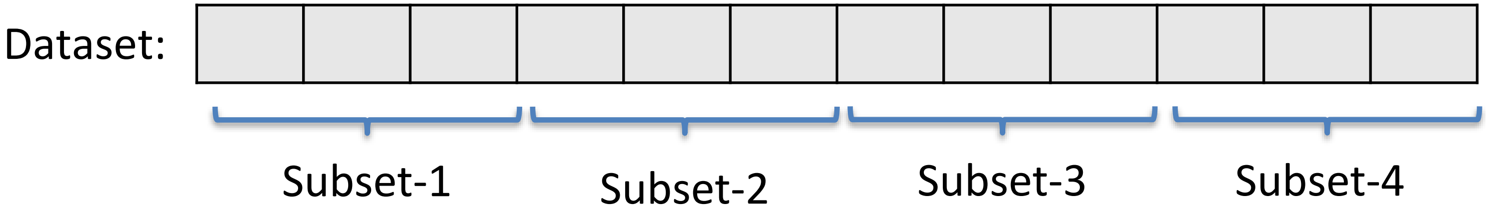


K-fold Cross Validation

- **Problem:** getting *ground truth* data expensive
- **Problem:** need different test data for each test

- **Goal:** minimize training+test data needed
- **Idea:** split training data into K subsets; use K-1 for *training* and one for *development testing*
- Repeat K times and average performance
- Common K values are 5 and 10
- Best practice: hold out a <u>final</u> test data set

K-fold Cross Validation



Iteration 1:	Training: Subset-1,2,3	Test:Subset-4
Iteration 2:	Training: Subset-1,3,4	Test:Subset-2
Iteration 3:	Training: Subset-2,4,1	Test:Subset-3

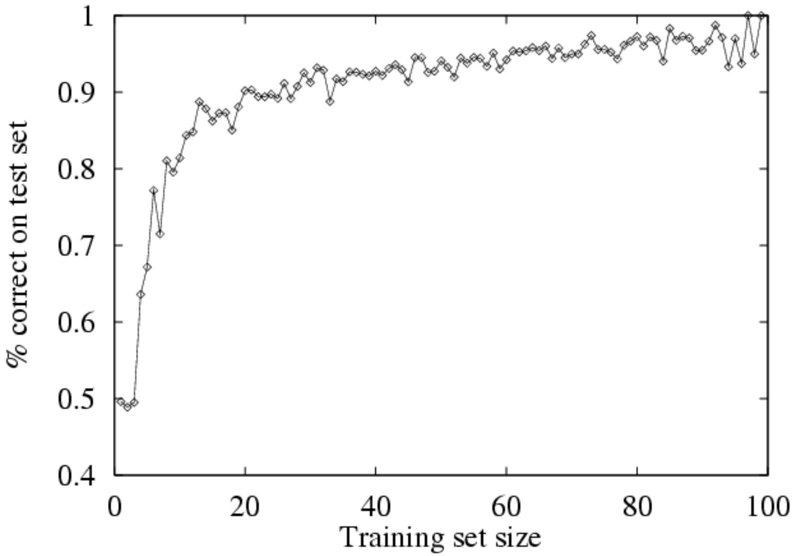
Leave one out

- K-fold cross validation can be too pessimistic, since it only trains with 80% or 90% of the data
- The leave one out evaluation is an alternative

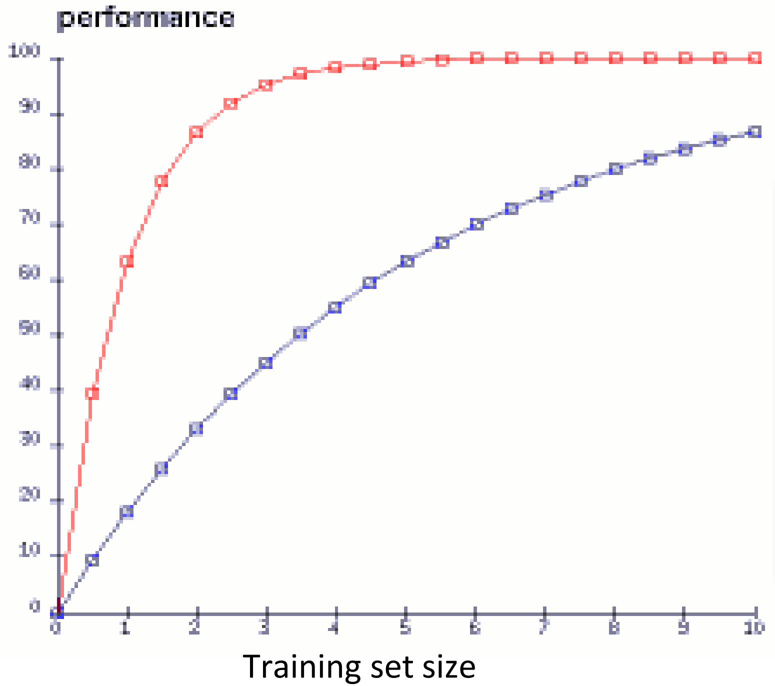
- 1 test sample taken at a time
- K-fold cross validation where k = number of samples

Learning curve (1)

A learning curve shows accuracy (% correct) on test set as a function of training set size or (for neural networks) running time



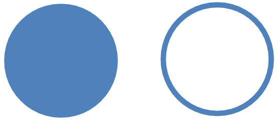
- When evaluating ML algorithms, steeper learning curves are better
- Represent faster learning with less data

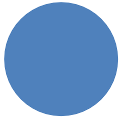


System with the red curve is better since it requires less data to achieve a given accuracy

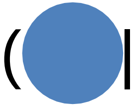
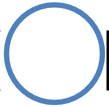
Classification Evaluation: the 2-by-2 contingency table

Let's assume there are two classes/labels



Assume  is the “positive” label

Given X, our classifier predicts either label

p(|X) vs. p(|X)

Classification Evaluation: the 2-by-2 contingency table

	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed		
Not selected/ not guessed		

Classes/Choices

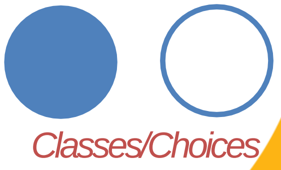
Classification Evaluation: the 2-by-2 contingency table

	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive(TP)	
Not selected/ not guessed		

Classes/Choices

Classification Evaluation: the 2-by-2 contingency table

	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive(TP)	False Positive(FP)
Not selected/ not guessed		



Classification Evaluation: the 2-by-2 contingency table

	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive(TP)	False Positive(FP)
Not selected/ not guessed	False Negative(FN)	

Classes/Choices

Classification Evaluation: the 2-by-2 contingency table

	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive(TP)	False Positive(FP)
Not selected/ not guessed	False Negative(FN)	True Negative(TN)

Classes/Choices

Classification Evaluation: the 2-by-2 contingency table

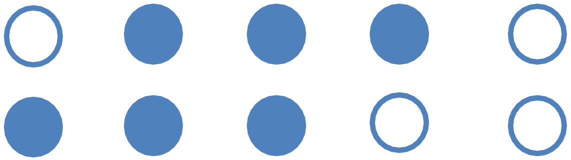
	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive(TP)	False Positive(FP)
Not selected/ not guessed	False Negative(FN)	True Negative(TN)

Classes/Choices

Construct this table by *counting* the number of TPs, FPs, FNs, TNs

Predicted:

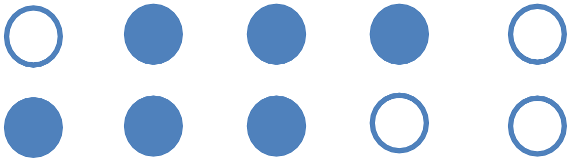
Actual:



Contingency Table Example

Predicted:

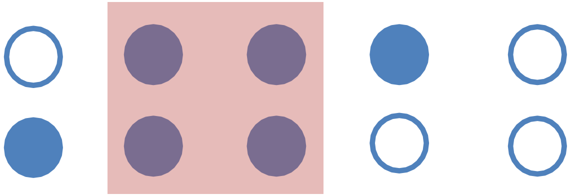
Actual:



	What is the actual label?	
What label does our system predict? (↓)	**Actually Correct**	**Actually Incorrect**
Selected/ Guessed	True Positive (TP)	False Positive (FP)
Not selected/ not guessed	False Negative (FN)	True Negative (TN)

Predicted:

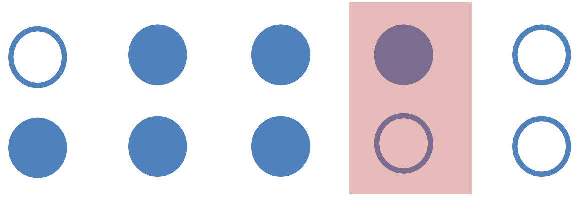
Actual:



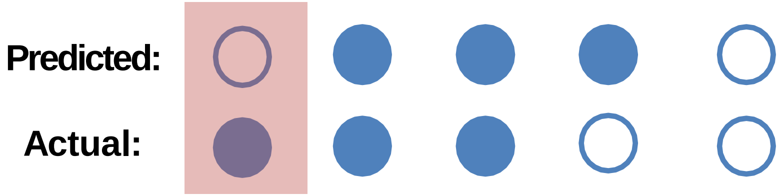
	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive (TP)= 2	False Positive (FP)
Not selected/ not guessed	False Negative (FN)	True Negative (TN)

Predicted:

Actual:



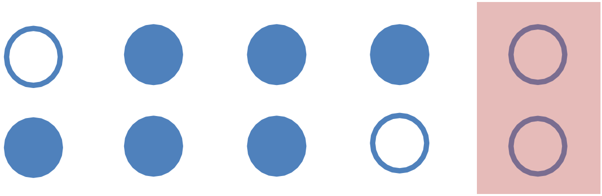
	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive (TP)= 2	False Positive (FP)=1
Not selected/ not guessed	False Negative (FN)	True Negative (TN)



	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive (TP)= 2	False Positive (FP)=1
Not selected/ not guessed	False Negative (FN)=1	True Negative (TN)

Predicted:

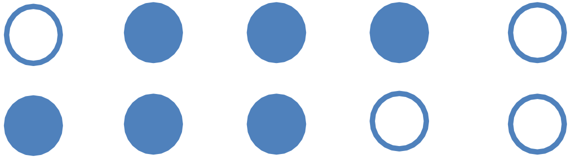
Actual:



	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive (TP)= 2	False Positive (FP)=1
Not selected/ not guessed	False Negative (FN)=1	True Negative (TN)=1

Predicted:

Actual:



	What is the actual label?	
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
Selected/ Guessed	True Positive (TP)= 2	False Positive (FP)=1
Not selected/ not guessed	False Negative (FN)=1	True Negative (TN)=1

Accuracy, Precision, and Recall

Accuracy: % of items correct

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

	Actually Correct	Actually Incorrect
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

Accuracy, Precision, and Recall

Accuracy: % of items correct

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

Precision: % of selected items that are correct

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

	Actually Correct	Actually Incorrect
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

Accuracy, Precision, and Recall

Accuracy: % of items correct

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

Precision: % of selected items that are correct

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

Recall: % of correct items that are selected

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

	Actually Correct	Actually Incorrect
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

Accuracy, Precision, and Recall

Accuracy: % of items correct

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

Precision: % of selected items that are correct

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

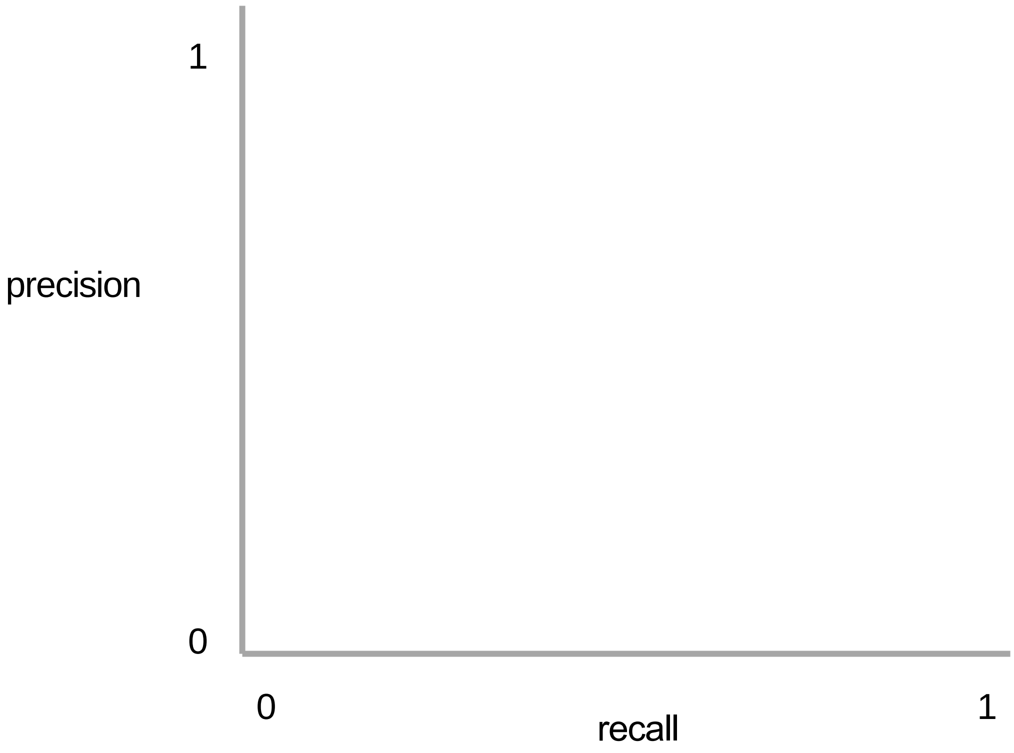
Min: 0 ☹
Max: 1 😀

Recall: % of correct items that are selected

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

	Actually Correct	Actually Incorrect
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

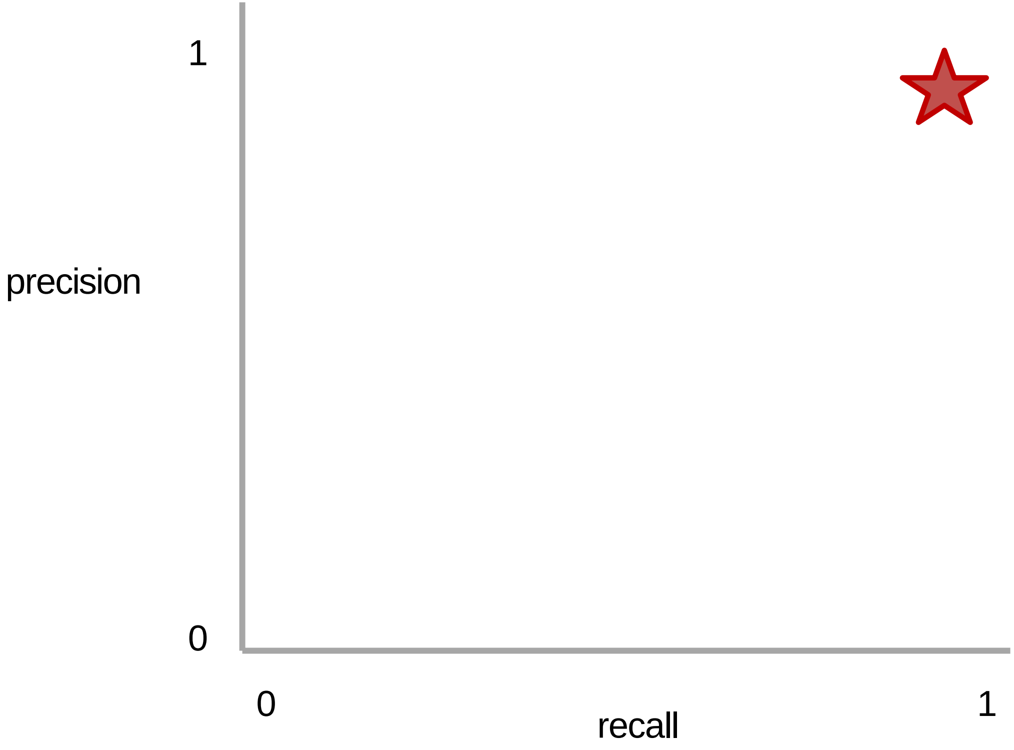
Precision and Recall Present a Tradeoff



model?

Q: Where do you want your ideal model

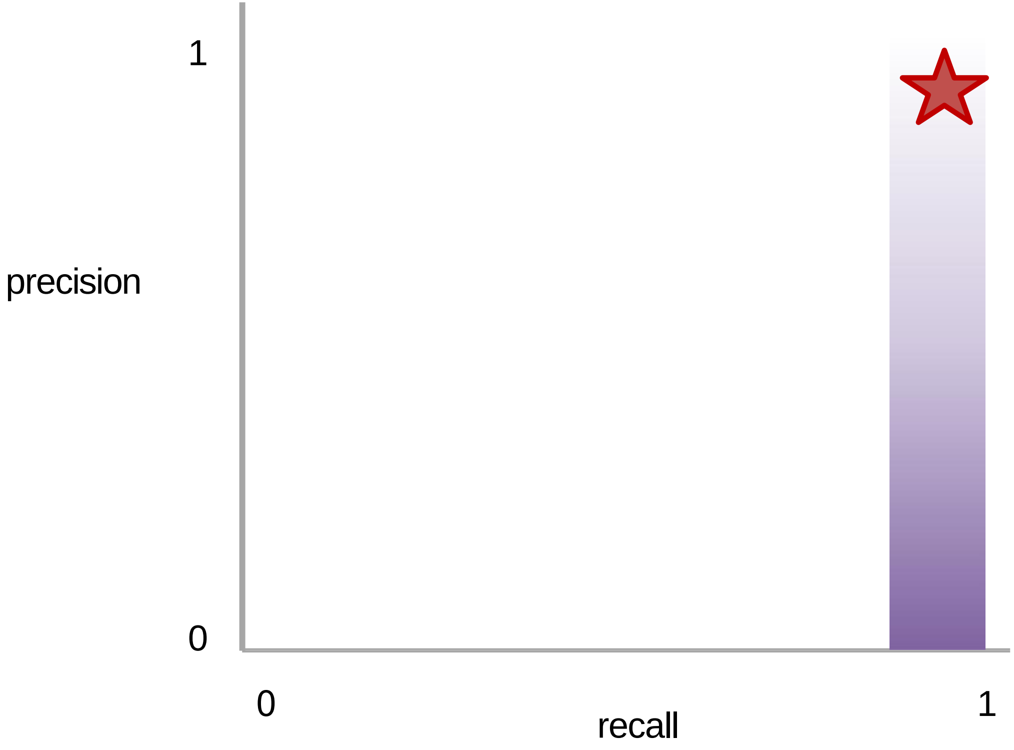
Precision and Recall Present a Tradeoff



Q: Where do you want your ideal model ?

Q: You have a model That identifies all correct instances. Where on this graph is it?

Precision and Recall Present a Tradeoff



Q: Where do you want your ideal model?

Q: You have a model That identifies all correct instances. Where on this graph is it?

Q: You have a model That identifies only correct instances. Where on this graph is it?

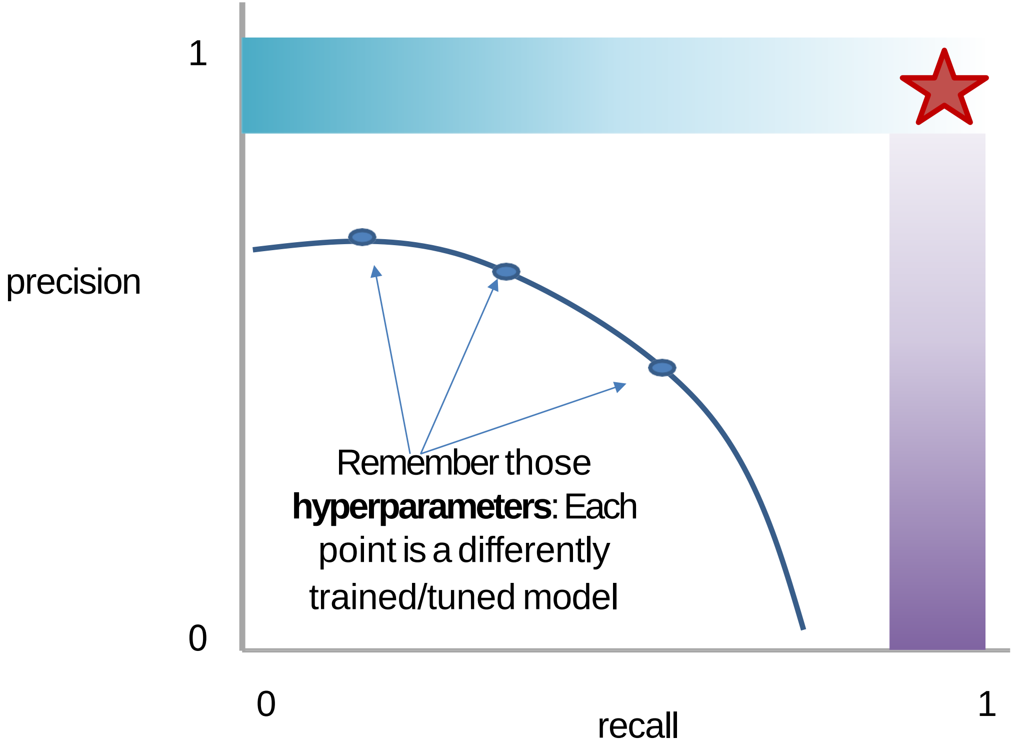
Precision and Recall Present a Tradeoff

Q: Where do you want your ideal model?

Q: You have a model That identifies all correct instances. Where on this graph is it?

Q: You have a model That identifies only correct instances. Where on this graph is it?

Precision and Recall Present a Tradeoff



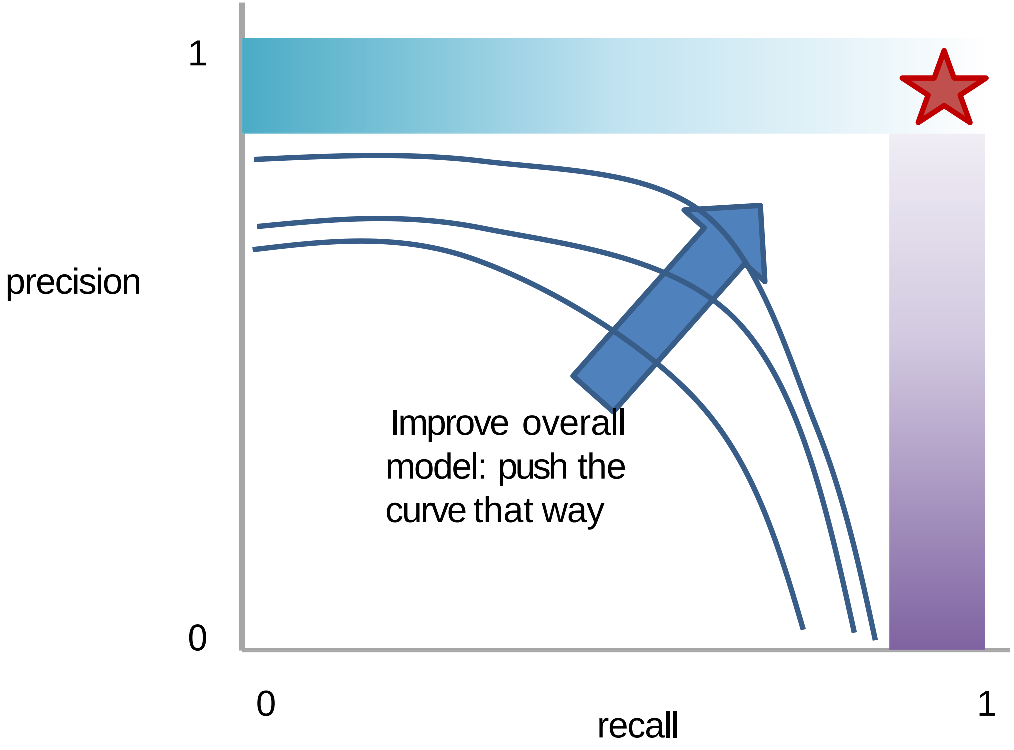
Q: Where do you want your ideal model ?

Q: You have a model That identifies all correct instances. Where on this graph is it?

Q: You have a model That identifies only correct instances. Where on this graph is it?

Idea: measure the tradeoff between precision and recall

Precision and Recall Present a Tradeoff



Q: Where do you want your ideal model ?

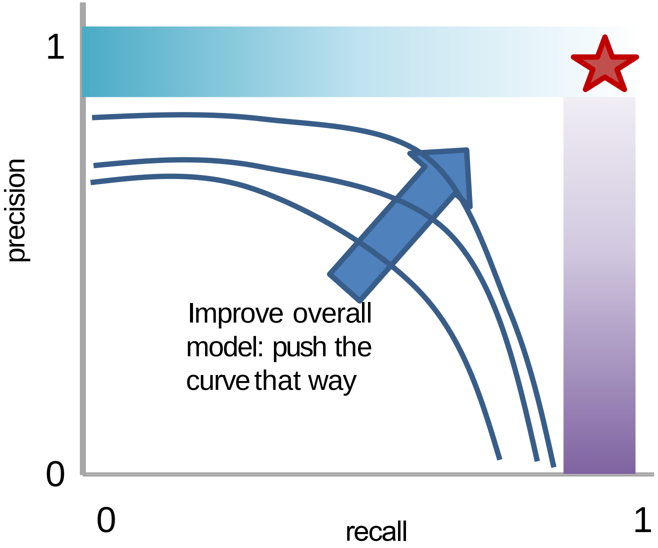
Q: You have a model That identifies all correct instances. Where on this graph is it?

Q: You have a model That identifies only correct instances. Where on this graph is it?

Idea: measure the tradeoff between precision and recall

Measure this Tradeoff:

Area Under the Curve (AUC)

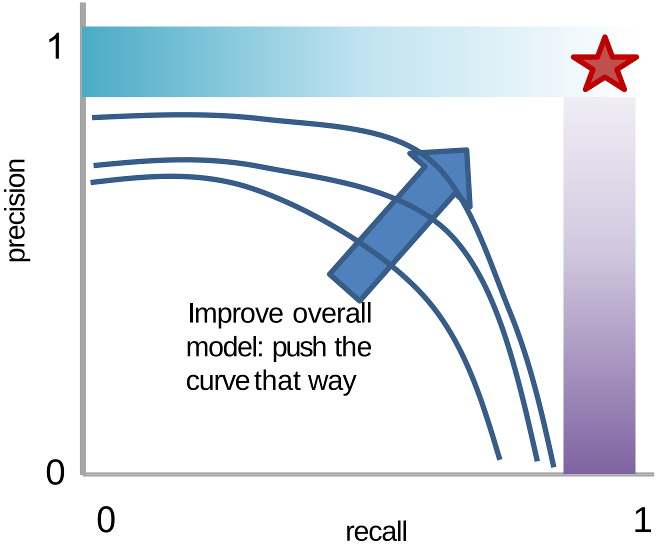


AUC measures the area under this trade-off curve

Min AUC: 0 ☹

Max AUC: 1 😀

Area Under the Curve (AUC)



Min AUC: 0 ☹
Max AUC: 1 😃

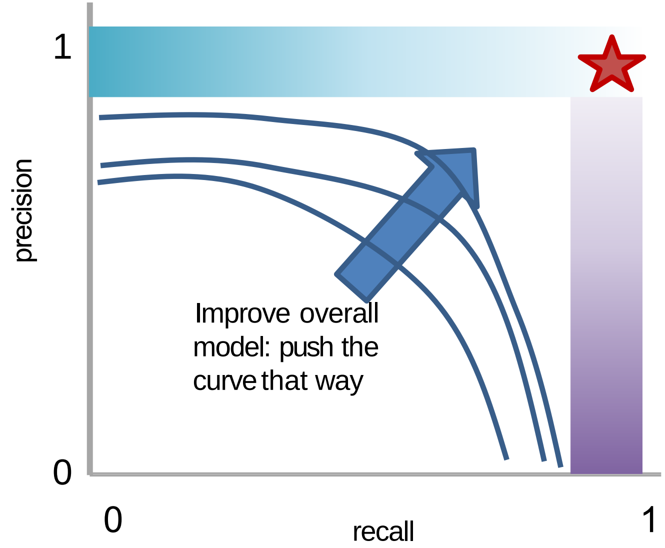
AUC measures the area under this trade-off curve

1. Computing the curve

   You need true labels & predicted labels with some score/confidence estimate

   Threshold the scores and for each threshold compute precision and recall

Area Under the Curve (AUC)



Min AUC: 0 ☹
Max AUC: 1 ☺

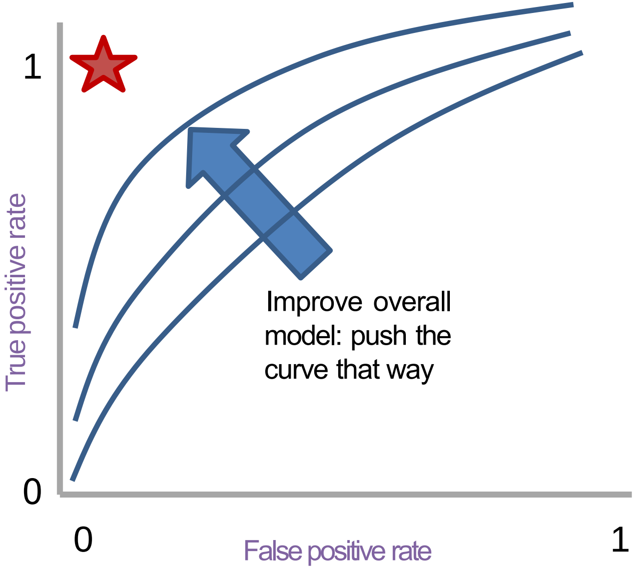
AUC measures the area under this tradeoff curve

1. Computing the curve
   You need true labels & predicted labels with some score/confidence estimate
   Threshold the scores and for each threshold compute precision and recall
2. Finding the area
   How to implement: trapezoidal rule (& others)

In practice: external library like the sklearn.metrics module

Measure A Slightly Different Tradeoff

ROC-AUC



Min ROC-AUC: 0.5 ☹
Max ROC-AUC: 1 😃

AUC measures the area under this tradeoff curve

1. Computing the curve
   - You need true labels & predicted labels with some score/confidence estimate
   - Threshold the scores and for each threshold compute metrics
2. Finding the area
   - How to implement: trapezoidal rule (& others)

In practice: external library like the sklearn.metrics module

Main variant: ROC-AUC

Same idea as before but with some flipped metrics

A combined measure: F

Weighted (harmonic) average of **P**recision & **R**ecall

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}}$$

A combined measure: F

Weighted (harmonic) average of **P**recision & **R**ecall

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(1+\beta^2) * P * R}{(\beta^2 * P) + R}$$

algebra
(not important)

A combined measure: F

Weighted (harmonic) average of **P**recision & **R**ecall

$$F = \frac{(1+\beta^2) \; * P \; * R}{(\beta^2 * P) + R}$$

Balanced F1 measure: β=1

$$F_1 = \frac{2 \; * P \; * R}{P + R}$$

P/R/F in a Multi-class Setting: Micro- vs. Macro-Averaging

If we have more than one class, how do we combine multiple performance measures into one quantity?

Macroaveraging: Compute performance for each class, then average.

Microaveraging: Collect decisions for all classes, compute contingency table, evaluate.

Micro- vs. Macro-Averaging

Macroaveraging: Compute performance for each class, then average.

$$\text{macroprecision} = \sum_c \frac{\mathrm{TP_c}}{\mathrm{TP_c} + \mathrm{FP_c}} = \sum_c \text{precision}_c$$

(missing 1/C)

Microaveraging: Collect decisions for all classes, compute contingency table, evaluate.

$$\text{microprecision} = \frac{\sum_\mathrm{c} \mathrm{TP_c}}{\sum_\mathrm{c} \mathrm{TP_c} + \sum_\mathrm{c} \mathrm{FP_c}}$$

Micro- vs. Macro-Averaging: Example

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro Ave. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

Macroaveraged precision: (0.5 + 0.9)/2 = 0.7

Microaveraged precision: 100/120 = .83

Microaveraged score is dominated by score on frequent classes

Confusion Matrix: Generalizing the 2-by-2 contingency table

		Correct Value		
		●	○	▭
Guessed Value	●	#	#	#
	○	#	#	#
	▭	#	#	#

UMBC

Confusion Matrix: Generalizing the 2-by-2 contingency table

		Correct Value		
		●	○	▭
Guessed Value	●	80	9	11
	○	7	86	7
	▭	2	8	9

Q: Is this a good result?

Confusion Matrix: Generalizing the 2-by-2 contingency table

		Correct Value		
		●	○	▭
Guessed Value	●	30	40	30
	○	25	30	50
	▭	30	35	35

Q: Is this a good result?

Confusion Matrix: Generalizing the 2-by-2 contingency table

		Correct Value		
		●	○	▭
Guessed Value	●	7	3	90
	○	4	8	88
	▭	3	7	90

Q: Is this a good result?