Introduction to Machine Learning: Methodology and Classification Evaluation

Frank Ferraro – ferraro@umbc.edu

CMSC 473/673

Some slides adapted from 3SLP

Outline

Classification (Methodology)

Evaluation

Reminder!

ML/NLP Framework for Prediction

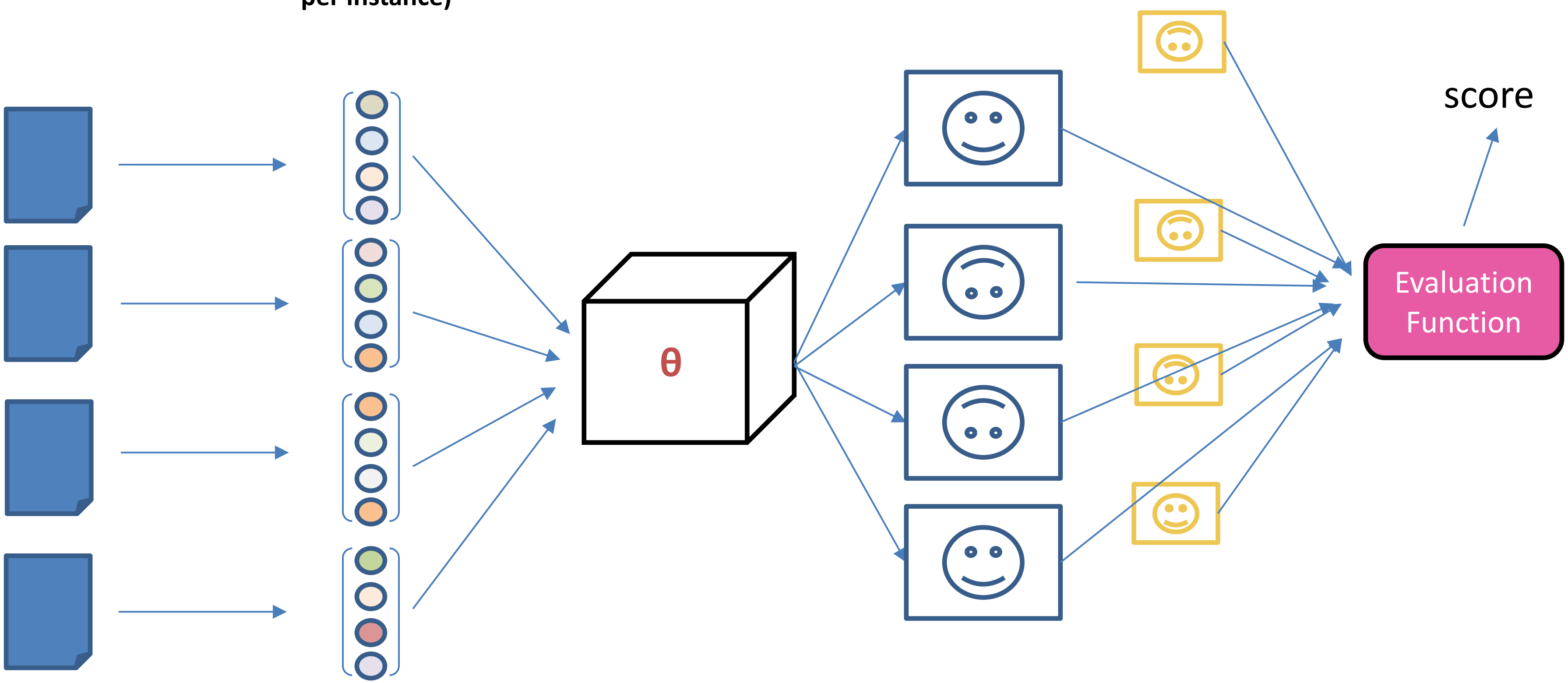
instances

features: **K-dimensional vector representations** (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ

"Gold" (correct) labels

Evaluation Function



Reminder!

$s = p_\theta($ Michael Jordan, coach Phil Jackson and the star cast, including Scottie Pippen, took the Chicago Bulls to six National Basketball Association championships. $)$

Goal: Learn parameters (weights) θ to develop a scoring function that says how "good" some provided text is

Classify with Uncertainty

$$\text{best label} = \arg\max_{\text{label}} P(\text{label}|\text{example})$$

*Use probabilities**

*There are non-probabilistic ways to handle uncertainty… but probabilities sure are handy!

Classification

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

POLITICS	.05
TERRORISM	.48
SPORTS	.0001
TECH	.39
HEALTH	.0001
FINANCE	.0002
...	

Source: http://www.nytimes.com/2016/09/20/nyregion/cellphone-alerts-used-in-search-of-manhattan-bombing-suspect.html

Reminder!

Classification Types (Terminology)

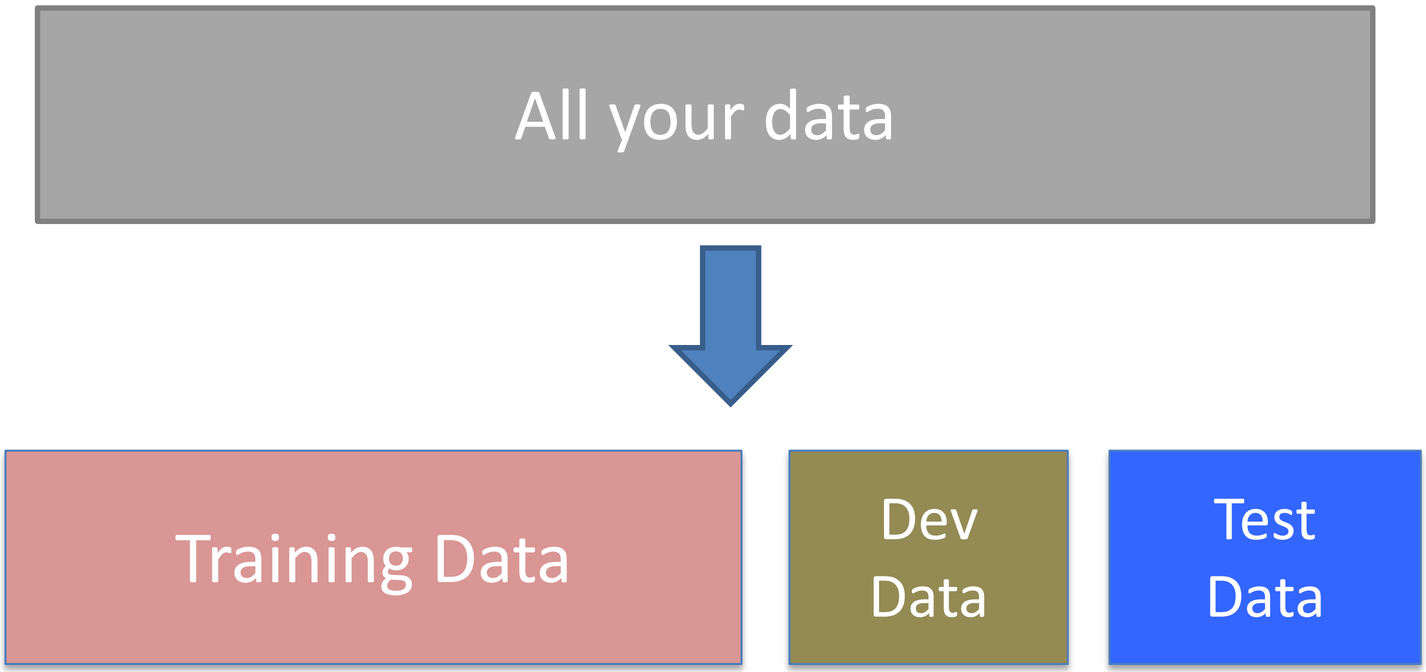
	Number of Tasks (Domains) Labels are Associated with	**# Label Types**	**Example**
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, …}
Multi-label Classification	1	> 2	Sentiment: Choose multiple of {positive, angry, sad, excited, …}
Multi-task Classification	> 1	Per task: 2 or > 2 (can apply to binary or multi-class)	Task 1: part-of-speech Task 2: named entity tagging … --------------------- Task 1: document labeling Task 2: sentiment

Outline

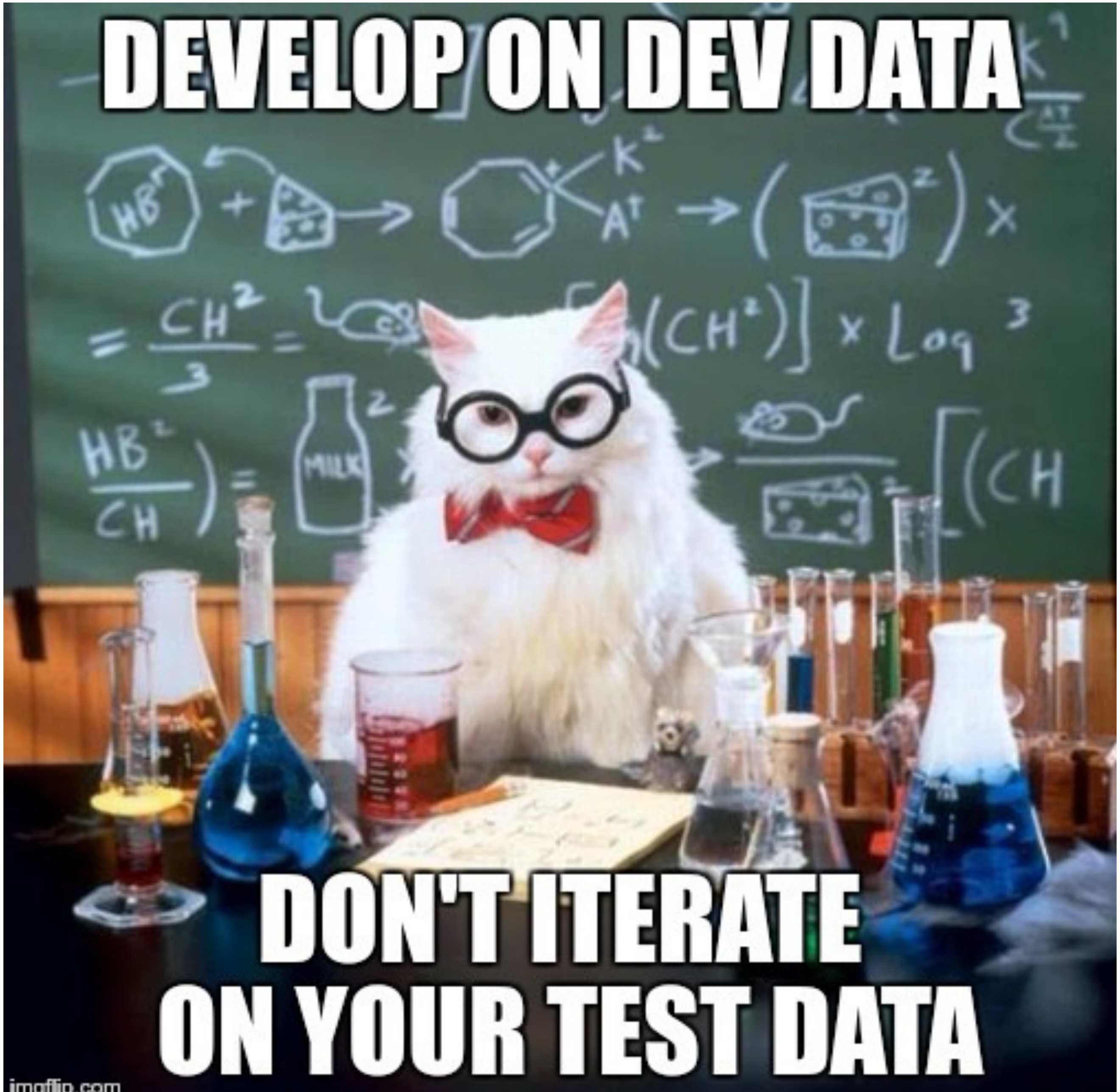
Classification (Methodology)

Evaluation

Experimenting with Machine Learning Models



Rule #1


DEVELOP ON DEV DATA
DON'T ITERATE
ON YOUR TEST DATA
imgflip.com

Experimenting with Machine Learning Models

What is "correct?"

What is working "well?"

Training Data

Dev Data

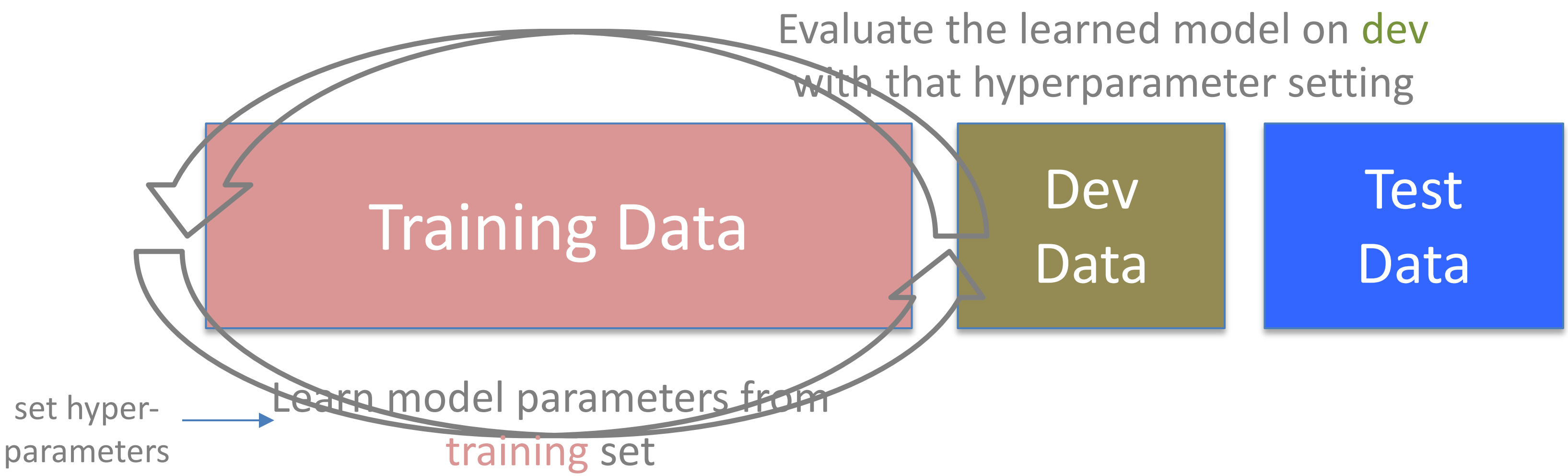
Test Data

set hyper-parameters → Learn model parameters from training set

Experimenting with Machine Learning Models

What is "correct?"

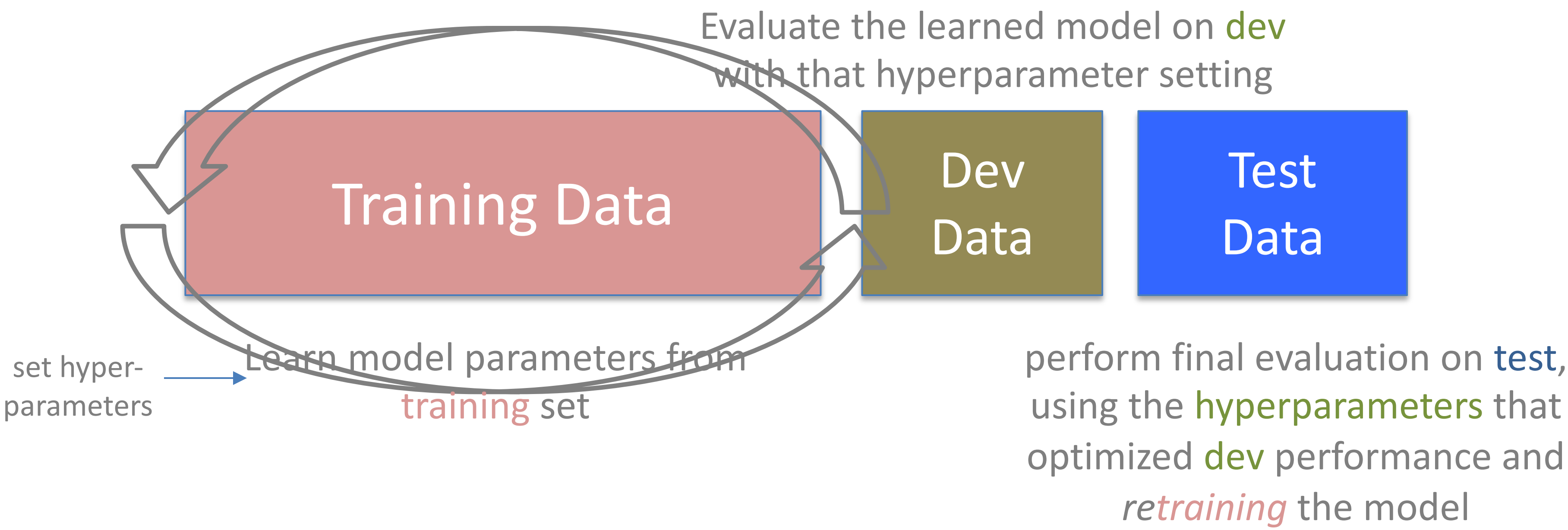
What is working "well?"



Experimenting with Machine Learning Models

What is "correct?"

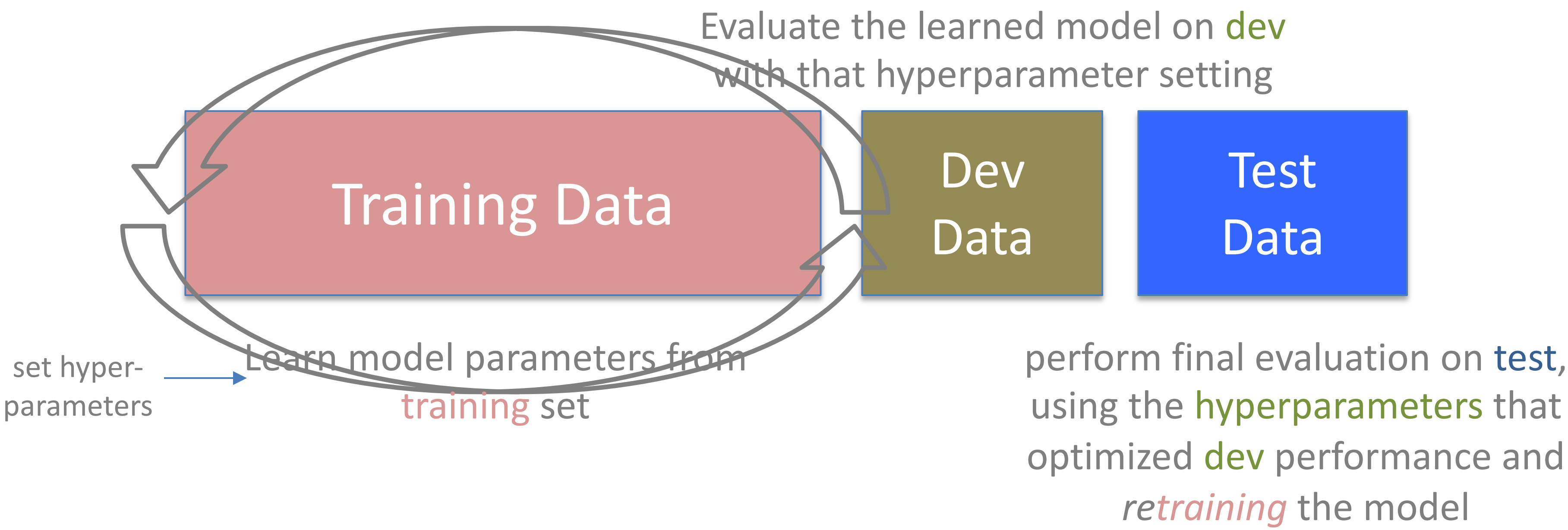
What is working "well?"



Experimenting with Machine Learning Models

What is "correct?"

What is working "well?"



Rule 1: DO NOT ITERATE ON THE TEST DATA

A Simplified Train-Dev-Test Cycle

```
train_split, dev_split, test_split = split_data(corpus)
best_score, best_hp = None, None
```

A Simplified Train-Dev-Test Cycle

```
train_split, dev_split, test_split = split_data(corpus)
best_score, best_hp = None, None
for hp in hyperparameter_config():
  model = make_model(hp)
  model.train(train_split)
```

A Simplified Train-Dev-Test Cycle

```
train_split, dev_split, test_split = split_data(corpus)
best_score, best_hp = None, None
for hp in hyperparameter_config():
  model = make_model(hp)
  model.train(train_split)
  score = model.evaluate(dev_split)
  if score > best_score:
    best_score = score
    best_hp = hp
```

A Simplified Train-Dev-Test Cycle

```
train_split, dev_split, test_split = split_data(corpus)
best_score, best_hp = None, None
for hp in hyperparameter_config():
  model = make_model(hp)
  model.train(train_split)
  score = model.evaluate(dev_split)
  if score > best_score:
    best_score = score
    best_hp = hp

best_model = make_model(best_hp)
best_model.train(train_split)
test_score = best_model.evaluate(test_split)
```

A More Realistic Train-Dev-Test Cycle

```
train_split, dev_split, test_split = split_data(corpus)

if is_training:
  best_score, best_hp = None, None
  for hp in hyperparameter_config():
    model = make_model(hp)
    model.train(train_split)
    score = model.evaluate(dev_split)
    if score > best_score:
      best_score = score
      best_hp = hp
      model.save_to_disk()
else:
  model = load_from_disk()
  test_score = model.evaluate(test_split)
```

A More Realistic Train-Dev-Test Cycle

```
train_split, dev_split, test_split = split_data(corpus)

if is_training:
  best_score, best_hp = None, None
  for hp in hyperparameter_config():
    model = make_model(hp)
    model.train(train_split)
    score = model.evaluate(dev_split)
    if score > best_score:
      best_score = score
      best_hp = hp
      model.save_to_disk()
else:
  model = load_from_disk()
  test_score = model.evaluate(test_split)
```

Split the training/dev and test cycles! (Training can sometimes take a while.)

A More Realistic Train-Dev-Test Cycle

```
train_split, dev_split, test_split =
split_data(corpus)

if is_training:
  best_score, best_hp = None, None
  for hp in hyperparameter_config():
    model = make_model(hp)
    model.train(train_split)
    score = model.evaluate(dev_split)
    if score > best_score:
      best_score = score
      best_hp = hp
      model.save_to_disk()
else:
  model = load_from_disk()
  test_score = model.evaluate(test_split)
```

- https://pytorch.org/tutorials/beginner/basics/optimization_tutorial.html
- https://pytorch.org/tutorials/beginner/basics/saveloadrun_tutorial.html

Central Question: How Well Are We Doing?

Classification
- Precision, Recall, F1
- Accuracy
- Log-loss
- ROC-AUC
- …

Regression
- (Root) Mean Square Error
- Mean Absolute Error
- …

Clustering
- Mutual Information
- V-score
- …

the ***task****: what* kind *of problem are you solving?*

Central Question: How Well Are We Doing?

Classification

- Precision, Recall, F1
- Accuracy
- Log-loss
- ROC-AUC
- …

Regression

- (Root) Mean Square Error
- Mean Absolute Error
- …

Clustering

- Mutual Information
- V-score
- …

the ***task****: what* kind *of problem are you solving?*

This does not have to be the same thing as the loss function you optimize

Training Loss vs. Evaluation Score

In training, compute loss to update parameters

Sometimes loss is a computational compromise

- surrogate loss

The loss you use might not be as informative as you'd like

Binary classification: 90 of 100 training examples are +1, 10 of 100 are -1

Some Classification Metrics

Accuracy

Precision
Recall

AUC (Area Under Curve)

F1

Confusion Matrix

Classification Evaluation: the 2-by-2 contingency table

Let’s assume there are two classes/labels

● ○

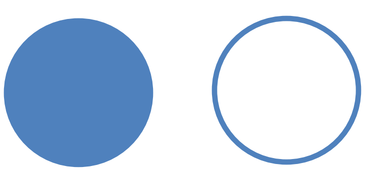
Assume ● is the “positive” label

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? (↓)	Actual Target Class ("●")	Not Target Class ("○")
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? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) *Actual* *Guessed*	
Not selected/ not guessed ("○")		

Classes/Choices

Classification Evaluation: the 2-by-2 contingency table

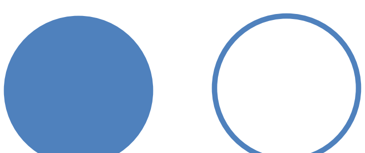
	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) ● *Actual* ● *Guessed*	False Positive (FP) ○ *Actual* ● *Guessed*
Not selected/ not guessed ("○")		

● ○

Classes/Choices

Classification Evaluation: the 2-by-2 contingency table

	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) ● *Actual* ● *Guessed*	False Positive (FP) ○ *Actual* ● *Guessed*
Not selected/ not guessed ("○")	False Negative (FN) ● *Actual* ○ *Guessed*	



Classes/Choices

Classification Evaluation: the 2-by-2 contingency table

	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) ● *Actual* ● *Guessed*	False Positive (FP) ○ *Actual* ● *Guessed*
Not selected/ not guessed ("○")	False Negative (FN) ● *Actual* ○ *Guessed*	True Negative (TN) ○ *Actual* ○ *Guessed*

● ○

Classes/Choices

Classification Evaluation: the 2-by-2 contingency table

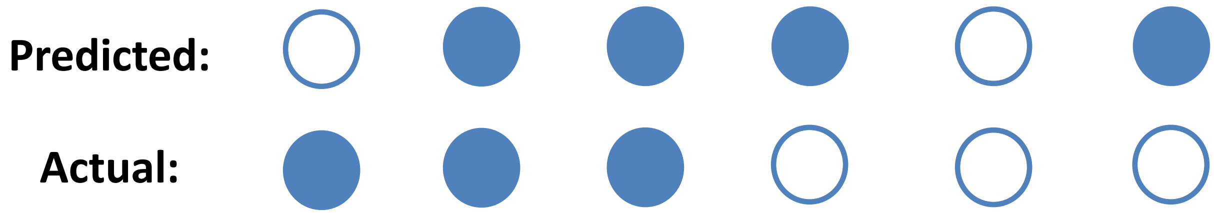
	What is the actual label?	
What label does our system predict? (↓)	**Actual Target Class ("●")**	**Not Target Class ("○")**
Selected/ Guessed ("●")	True Positive (TP) ● *Actual* ● *Guessed*	False Positive (FP) ○ *Actual* ● *Guessed*
Not selected/ not guessed ("○")	False Negative (FN) ● *Actual* ○ *Guessed*	True Negative (TN) ○ *Actual* ○ *Guessed*

● ○

Classes/Choices

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

Contingency Table Example



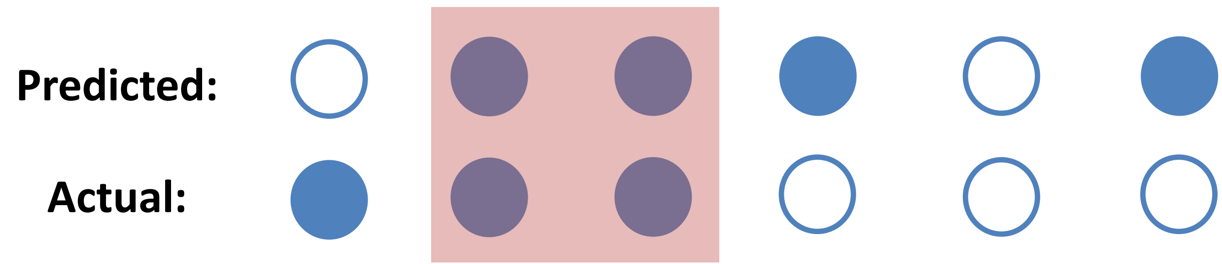
Contingency Table Example

Predicted: ○ ● ● ● ○ ●

Actual: ● ● ● ○ ○ ○

	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP)	False Positive (FP)
Not selected/ not guessed ("○")	False Negative (FN)	True Negative (TN)

Contingency Table Example

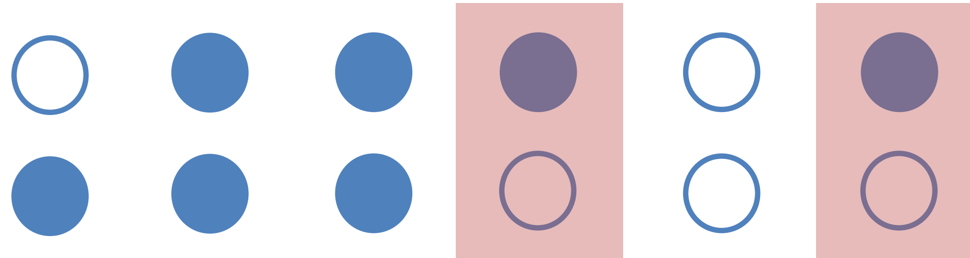


	What is the actual label?	
What label does our system predict? (↓)	**Actual Target Class ("●")**	**Not Target Class ("○")**
Selected/ Guessed ("●")	True Positive (TP) = 2	False Positive (FP)
Not selected/ not guessed ("○")	False Negative (FN)	True Negative (TN)

Contingency Table Example

Predicted:

Actual:

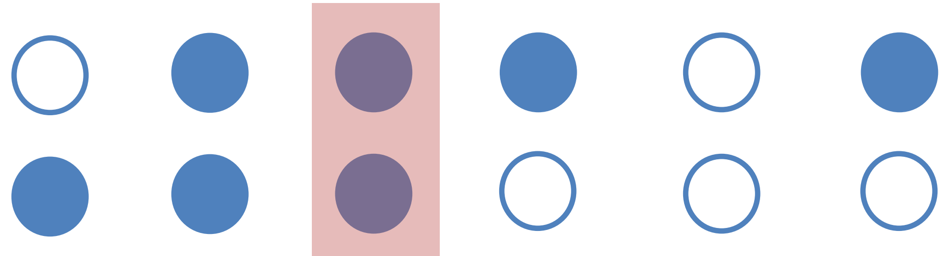


	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) = 2	False Positive (FP) = 2
Not selected/ not guessed ("○")	False Negative (FN)	True Negative (TN)

Contingency Table Example

Predicted:

Actual:

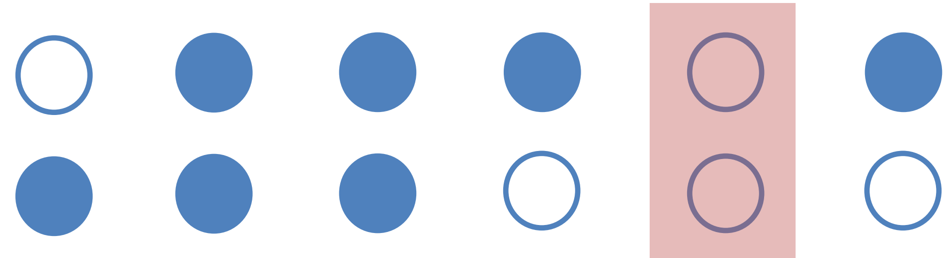


	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) = 2	False Positive (FP) = 2
Not selected/ not guessed ("○")	False Negative (FN) = 1	True Negative (TN)

Contingency Table Example

Predicted:

Actual:

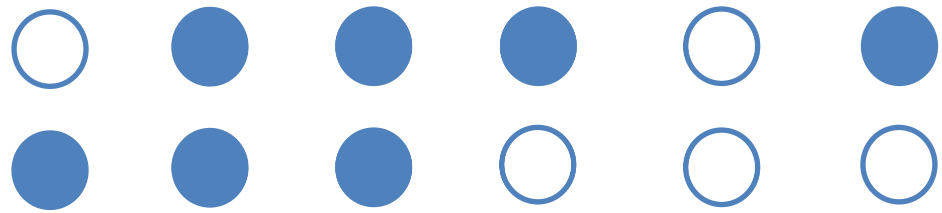


	What is the actual label?	
What label does our system predict? (↓)	**Actual Target Class ("●")**	**Not Target Class ("○")**
Selected/ Guessed ("●")	True Positive (TP) = 2	False Positive (FP) = 2
Not selected/ not guessed ("○")	False Negative (FN) = 1	True Negative (TN) = 1

Contingency Table Example

Predicted:

Actual:



	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) = 2	False Positive (FP) = 2
Not selected/ not guessed ("○")	False Negative (FN) = 1	True Negative (TN) = 1

Classification Evaluation: Accuracy, Precision, and Recall

Accuracy: % of items correct

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

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

Classification Evaluation: 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 Target	Actually Not Target
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

Classification Evaluation: 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 Target	Actually Not Target
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

Classification Evaluation: 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 😀

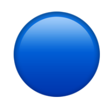
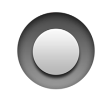
Recall: % of correct items that are selected

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

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

The Importance of “Polarity” in Binary Classification

Fundamentally: what are you trying to “identify” in your classification?

Are you trying to find  or ?

The Importance of "Polarity" in Binary Classification

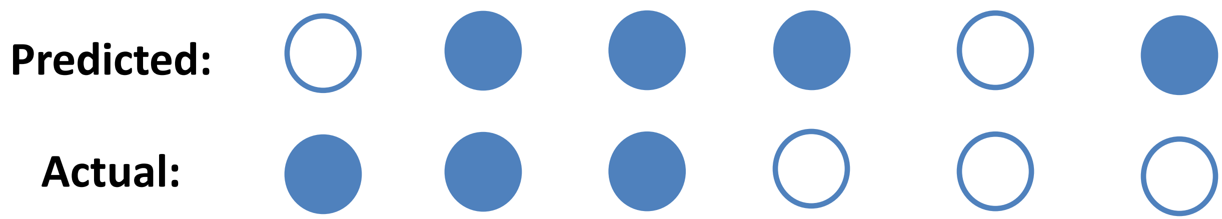
		Correct Value	
		●	○
Guessed Value	●	#	#
	○	#	#

Try to find 🔵: Where do the TP / FP / FN / FN values go?

The Importance of "Polarity" in Binary Classification

		Correct Value	
		●	○
Guessed Value	●	TP	FP
	○	FN	TN

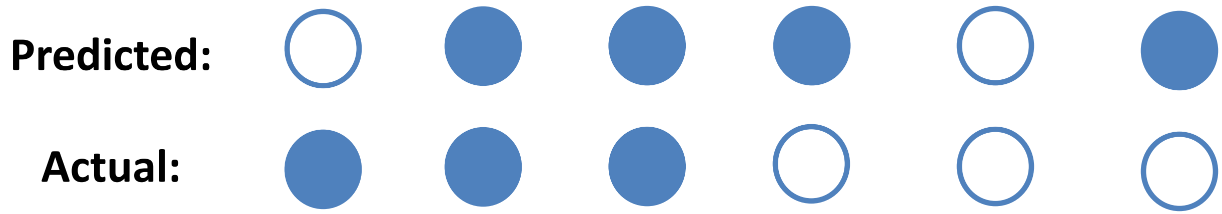
The Importance of "Polarity" in Binary Classification



		Correct Value	
Guessed Value		$TP = 2$	$FP = 2$
		$FN = 1$	$TN = 1$

What are the accuracy, recall, and precision values?

The Importance of “Polarity” in Binary Classification



		Correct Value	
Guessed Value		$TP = 2$	$FP = 2$
		$FN = 1$	$TN = 1$

What are the accuracy, recall, and precision values?

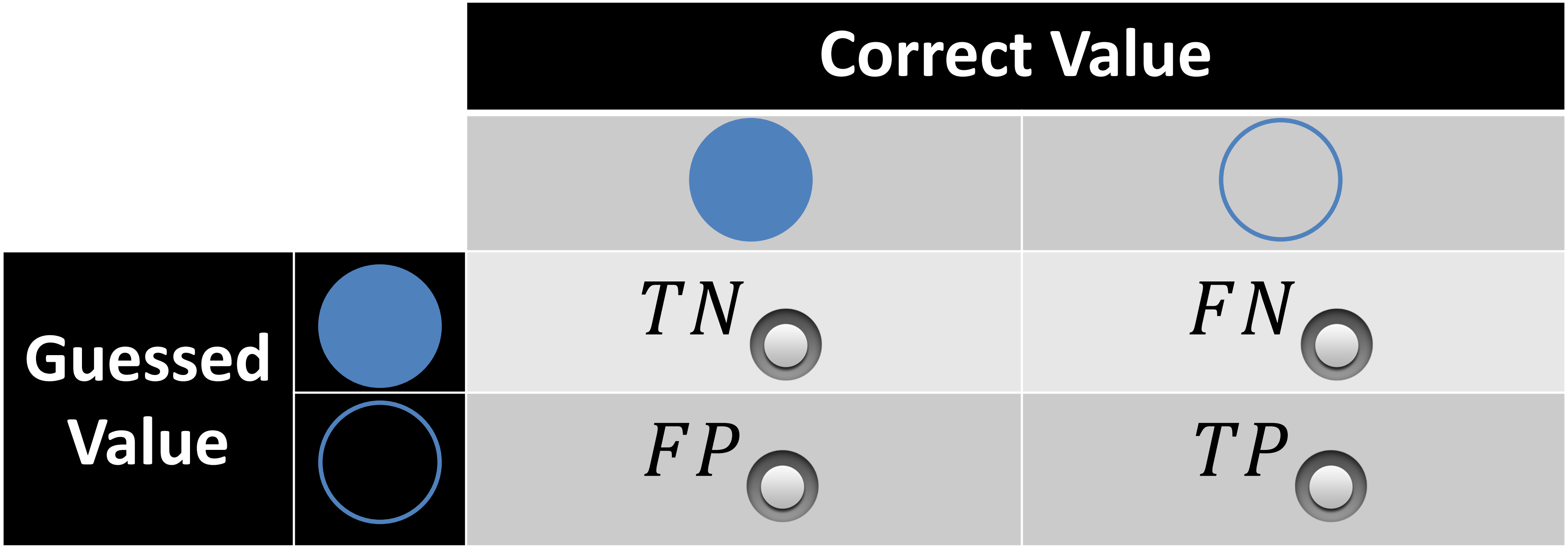
Accuracy: 50%
Recall: 66.67%
Precision: 50%

The Importance of “Polarity” in Binary Classification

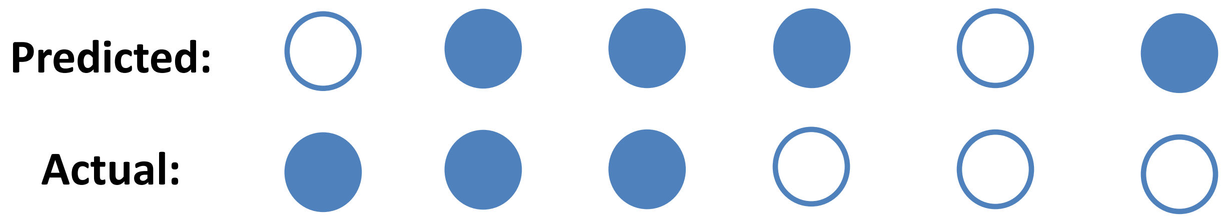
		Correct Value	
		●	○
Guessed Value	●	#	#
	○	#	#

Try to find : Where do the TP / FP / FN / FN values go?

The Importance of “Polarity” in Binary Classification



The Importance of "Polarity" in Binary Classification



		Correct Value	
		●	○
Guessed Value	●	$TN = 2$	$FN = 2$
	○	$FP = 1$	$TP = 1$

What are the accuracy, recall, and precision values?

The Importance of “Polarity” in Binary Classification

Predicted: ○ ● ● ● ○ ●

Actual: ● ● ● ○ ○ ○

		Correct Value	
		●	○
Guessed Value	●	$TN = 2$	$FN = 2$
	○	$FP = 1$	$TP = 1$

What are the accuracy, recall, and precision values?

Accuracy: 50%
Recall: 33.34%
Precision: 50%

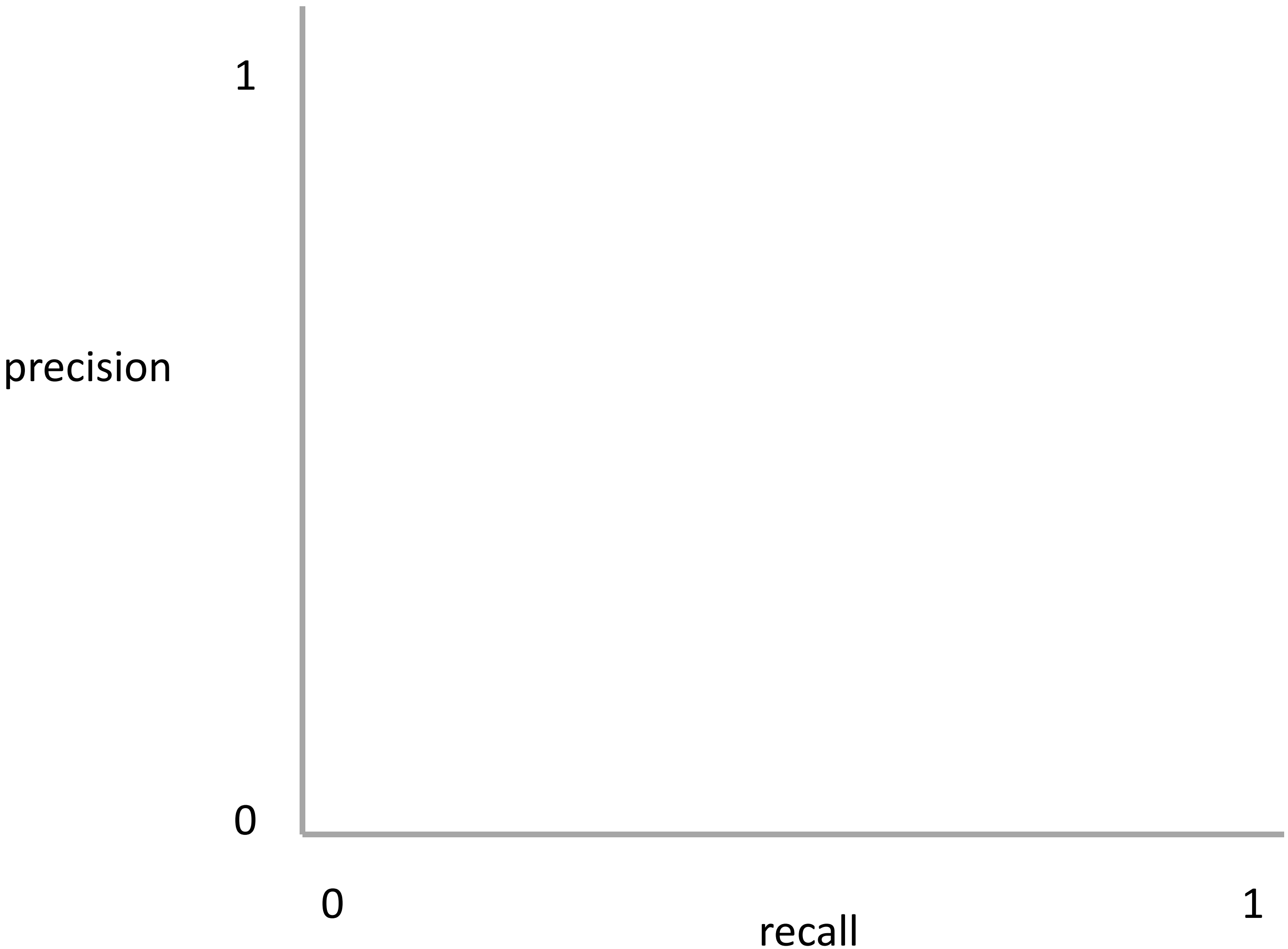
The Importance of "Polarity" in Binary Classification

		Correct Value	
		●	○
Guessed Value	●	$TP_{\bullet} = TN_{\circ}$	$FP_{\bullet} = FN_{\circ}$
	○	$FN_{\bullet} = FP_{\circ}$	$TN_{\bullet} = TP_{\circ}$

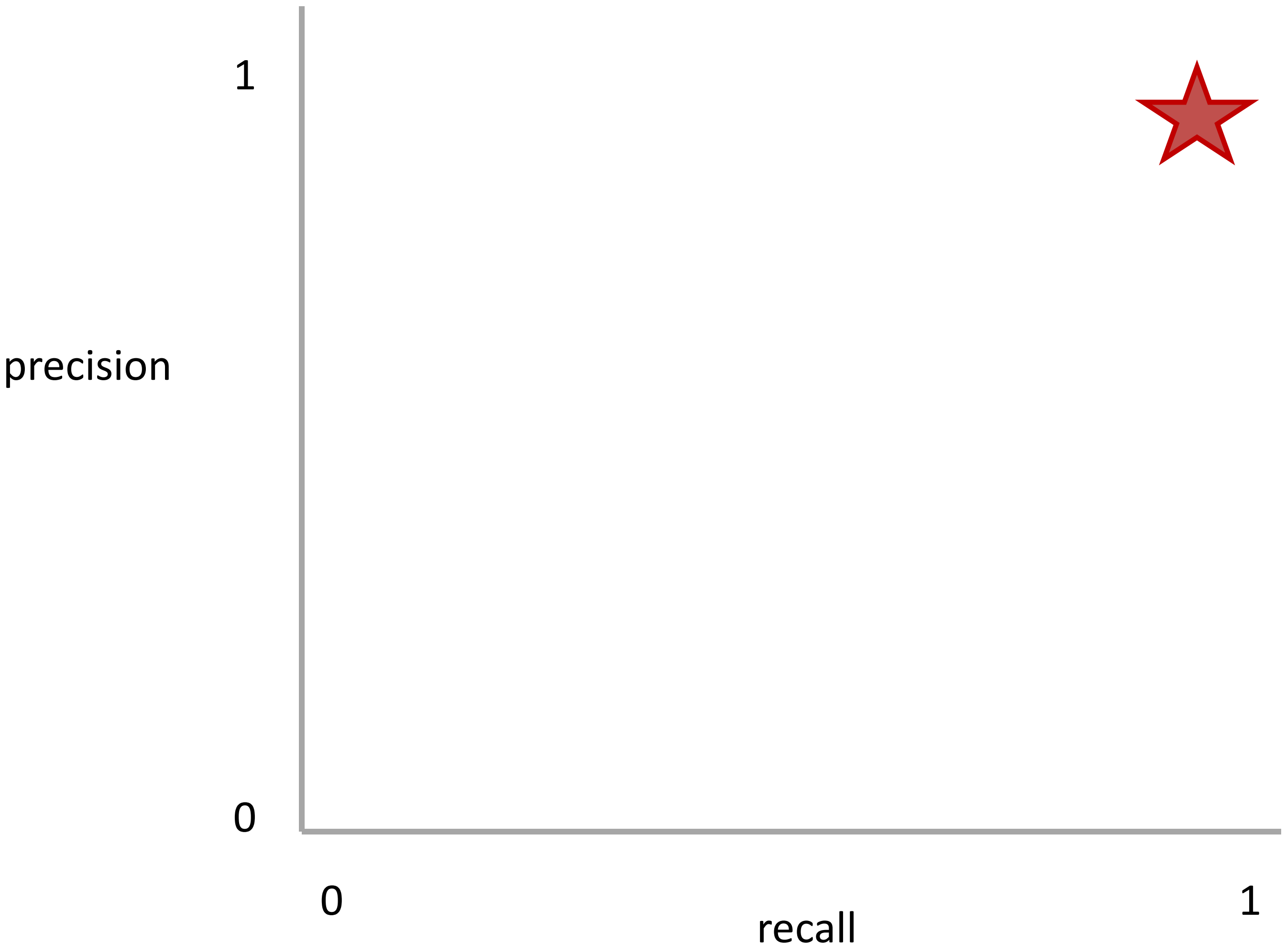
Remember: what are you trying to "identify" in your classification?

Precision and Recall Present a Tradeoff

Q: Where do you want your ideal model ?



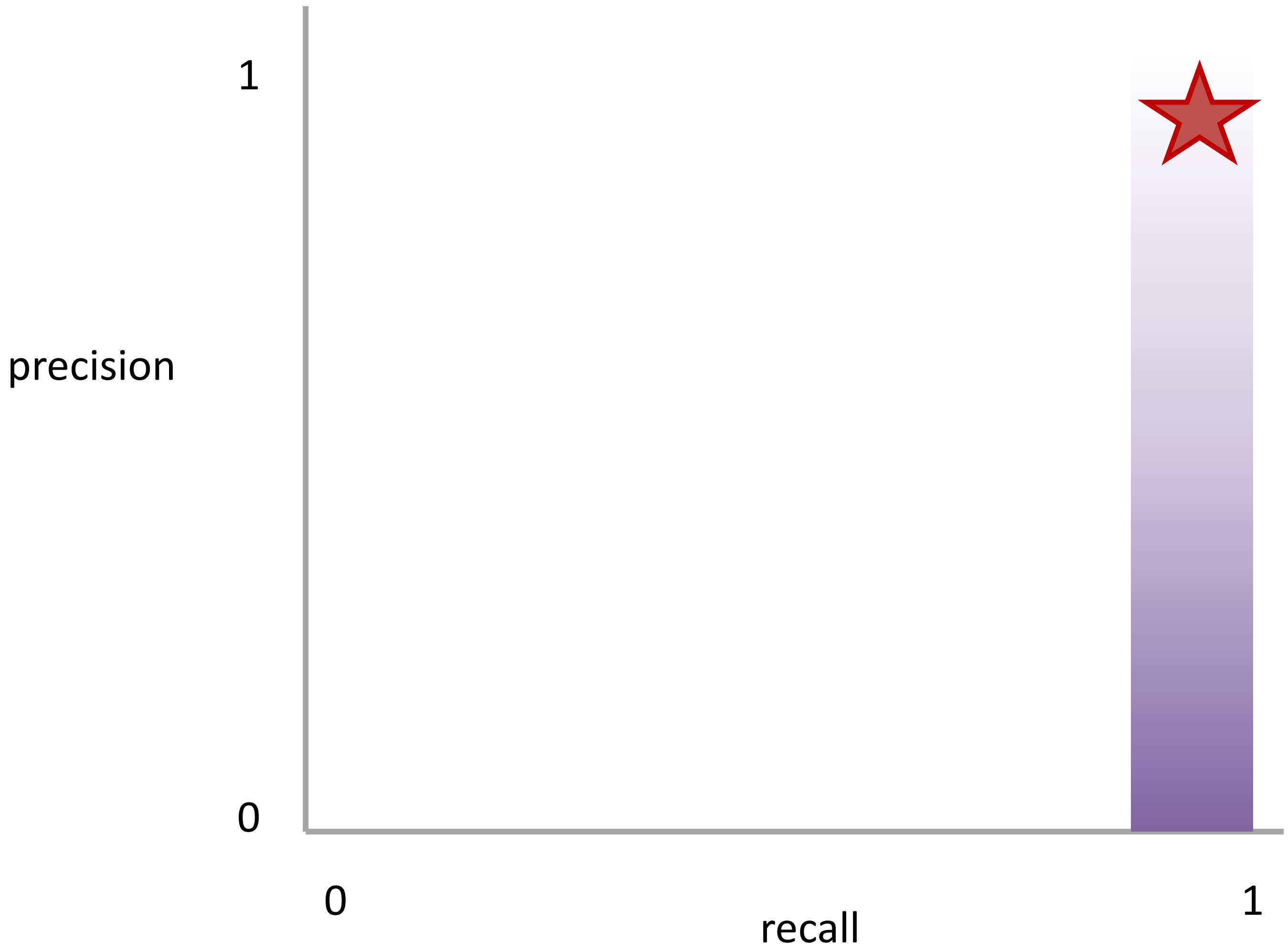
Precision and Recall Present a Tradeoff



Q: Where do you want your ideal model?

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

Precision and Recall Present a Tradeoff

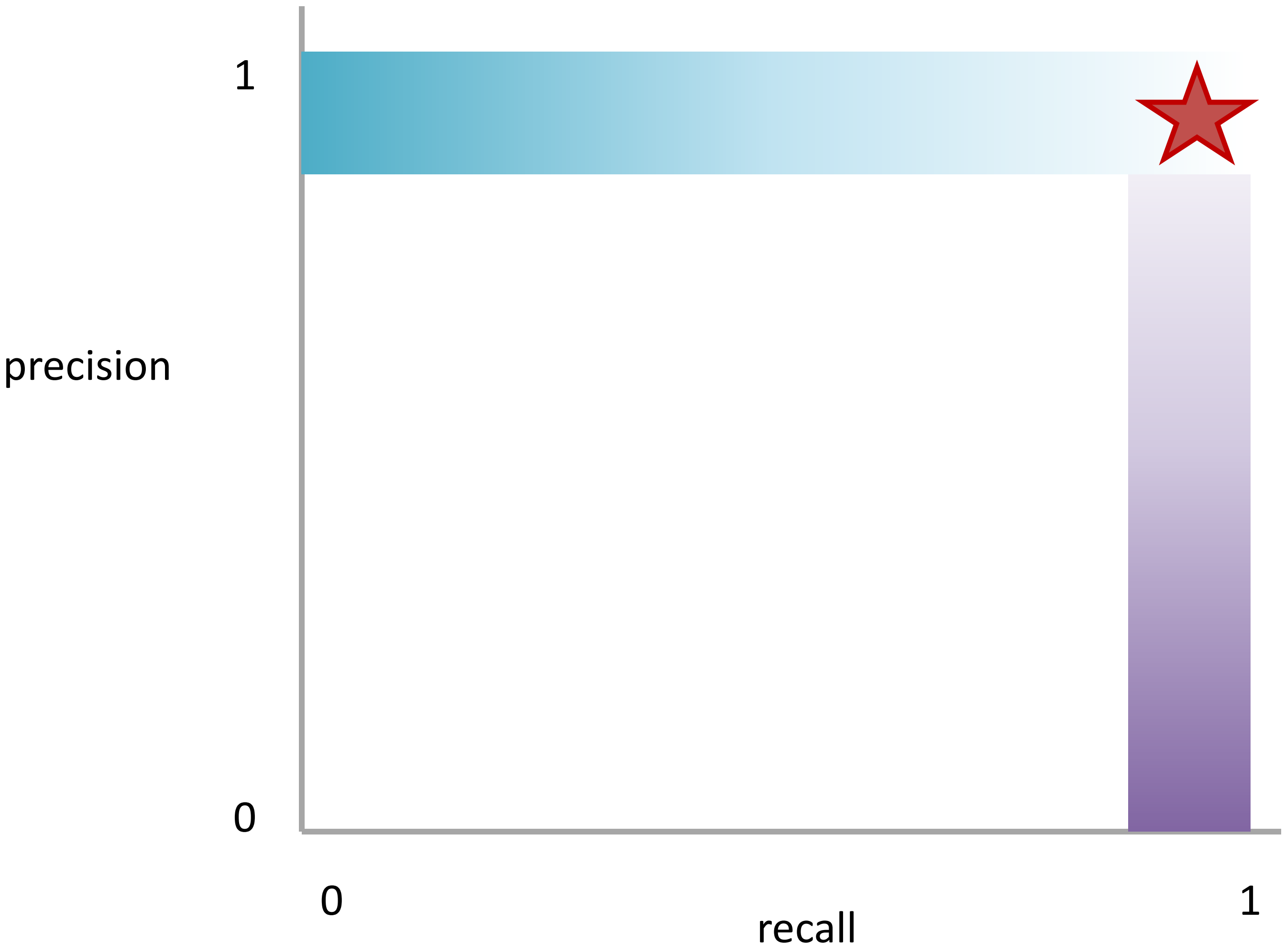


Q: Where do you want your ideal model?

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

Q: You have a model that only make correct predictions. Where on this graph is it?

Precision and Recall Present a Tradeoff

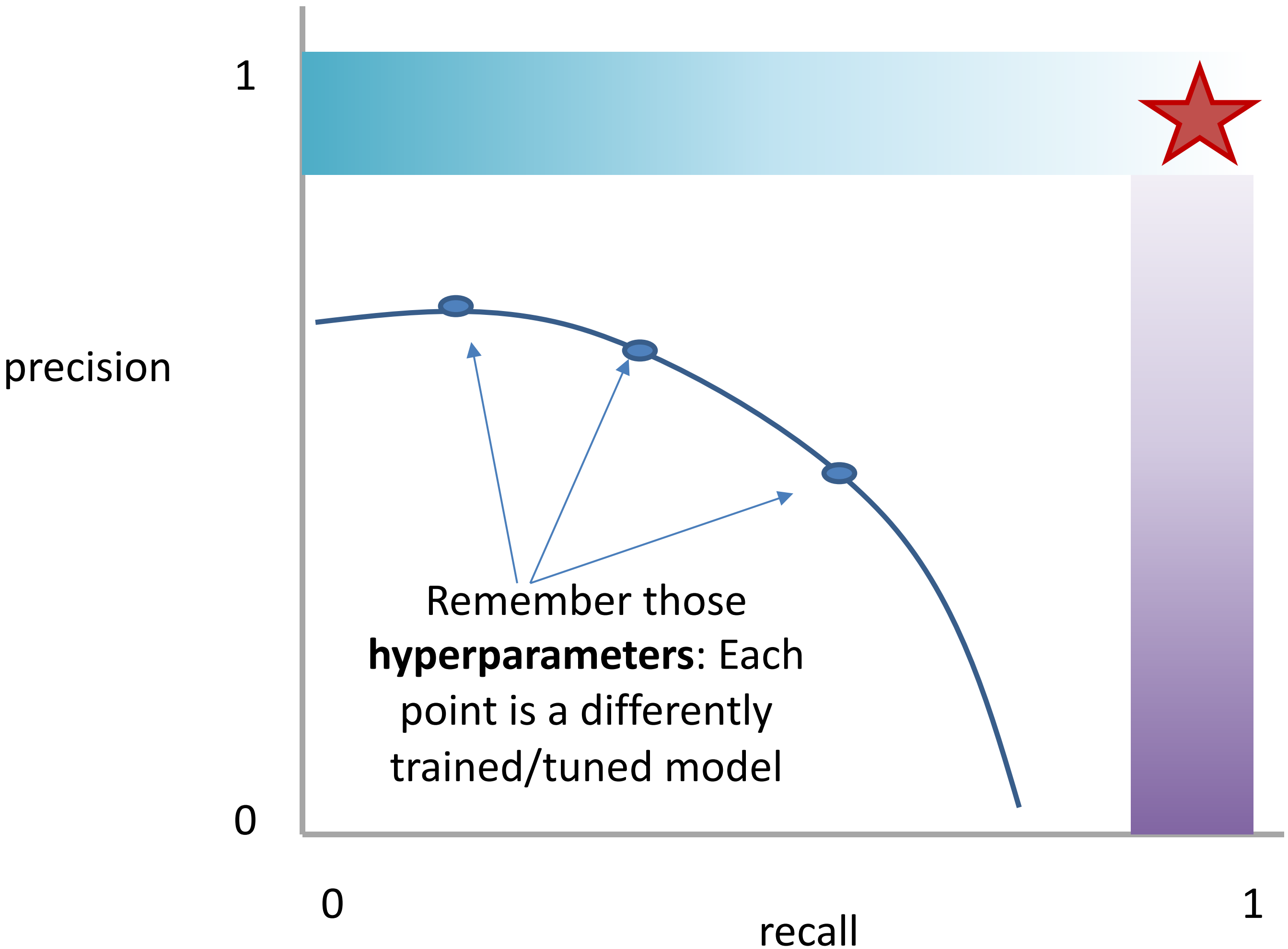


Q: Where do you want your ideal model ?

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

Q: You have a model that only make correct predictions. Where on this graph is it?

Precision and Recall Present a Tradeoff



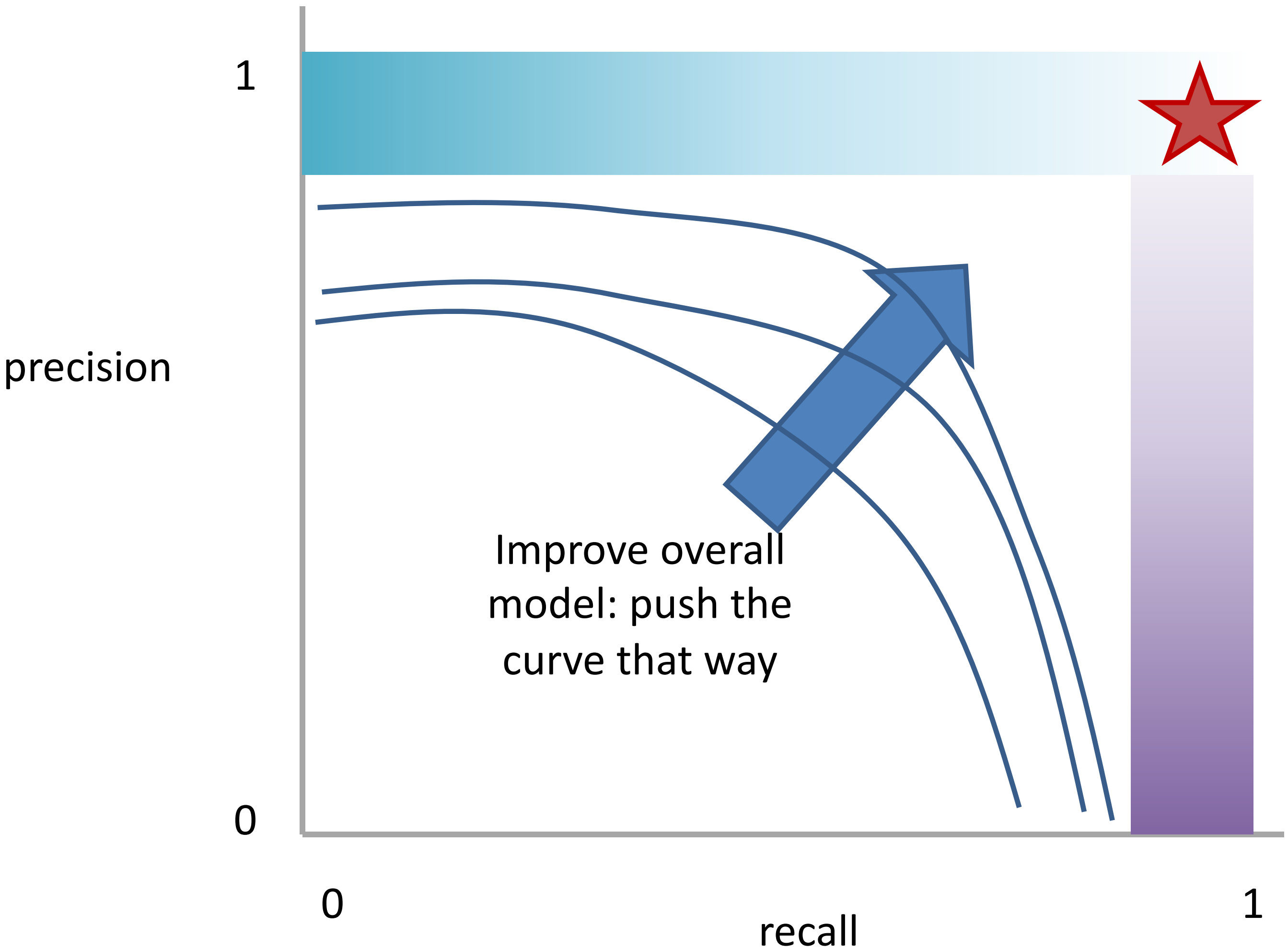
Q: Where do you want your ideal model ?

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

Q: You have a model that only make correct predictions. Where on this graph is it?

Idea: measure the tradeoff between precision and recall

Precision and Recall Present a Tradeoff



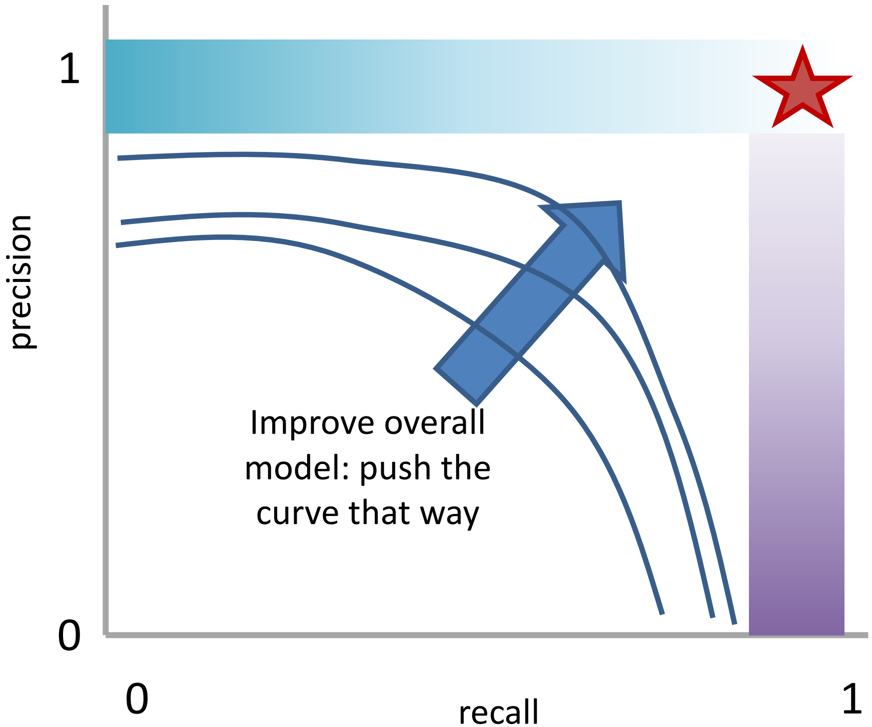
Q: Where do you want your ideal model ?

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

Q: You have a model that only make correct predictions. Where on this graph is it?

Idea: measure the tradeoff between precision and recall

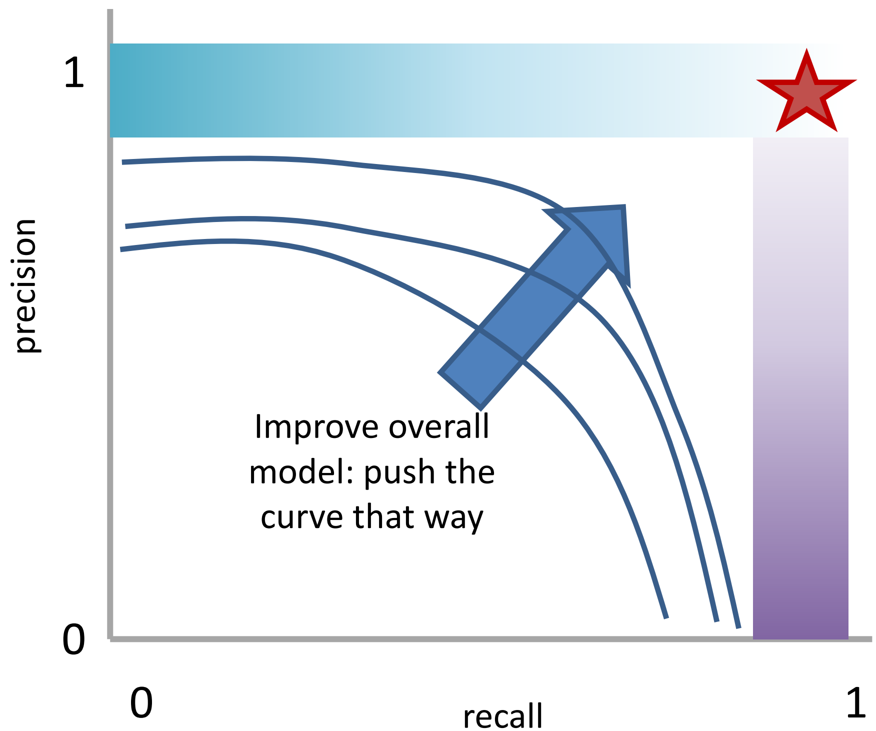
Measure this Tradeoff: Area Under the Curve (AUC)



AUC measures the area under this tradeoff curve

Min AUC: 0 ☹️
Max AUC: 1 😀

Measure this Tradeoff: Area Under the Curve (AUC)



Min AUC: 0 ☹
Max AUC: 1 😀

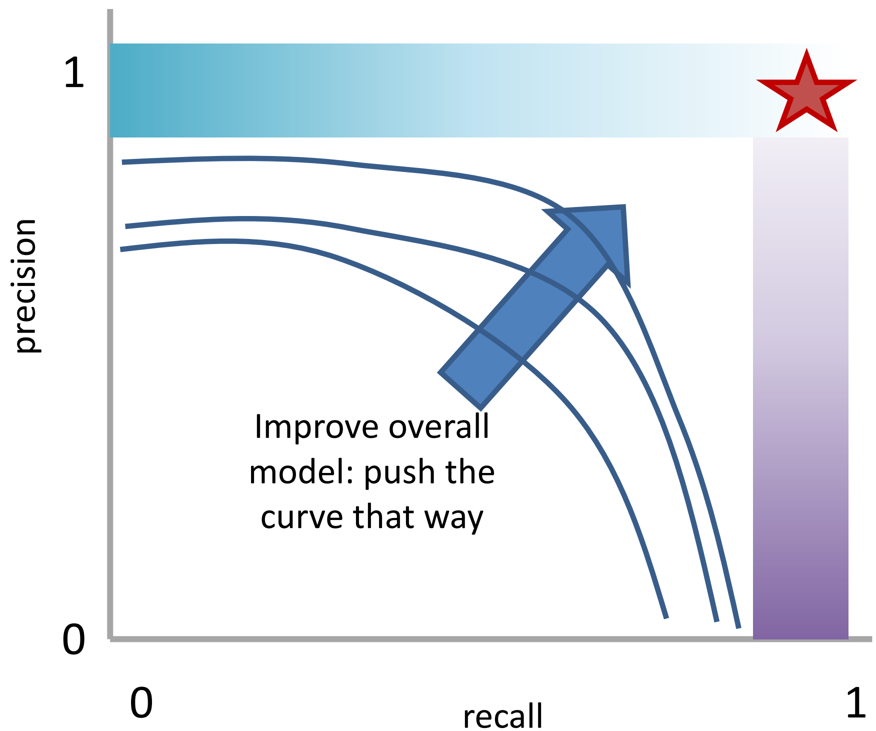
AUC measures the area under this tradeoff curve

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

Measure this Tradeoff: 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

A combined measure: F

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

F1 measure: equal weighting between precision and recall

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

A combined measure: F

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

F1 measure: equal weighting between precision and recall

$$F_1 = \frac{2 * P * R}{P + R} = \frac{2 * TP}{2 * TP + FP + FN}$$

(useful when $P = R = 0$)

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.

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

Macroaveraging: Compute performance for each class, then average.

$$\text{macroprecision} = \frac{1}{C}\sum_{c}\frac{\mathrm{TP_c}}{\mathrm{TP_c}+\mathrm{FP_c}} = \frac{1}{C}\sum_{c}\text{precision}_{c}$$

$$\text{macrorecall} = \frac{1}{C}\sum_{c}\frac{\mathrm{TP_c}}{\mathrm{TP_c}+\mathrm{FN_c}} = \frac{1}{C}\sum_{c}\text{recall}_{\mathrm{c}}$$

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

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

$$\text{microrecall} = \frac{\sum_{\mathrm{c}}\mathrm{TP_c}}{\sum_{\mathrm{c}}\mathrm{TP_c}+\sum_{\mathrm{c}}\mathrm{FN_c}}$$

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

Macroaveraging: Compute performance for each class, then average.

when to prefer macroaveraging?

$$\text{macroprecision} = \frac{1}{C}\sum_{c}\frac{\mathrm{TP_c}}{\mathrm{TP_c}+\mathrm{FP_c}} = \frac{1}{C}\sum_{c}\text{precision}_c$$

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

when to prefer microaveraging?

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

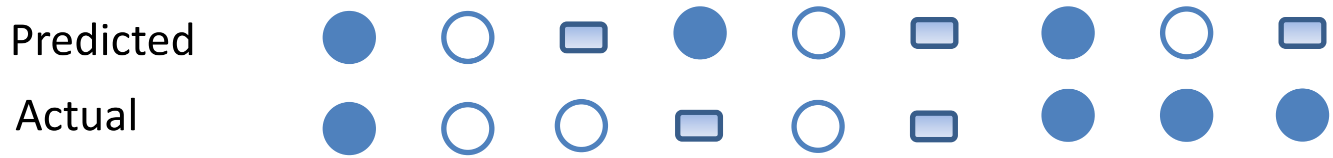
But how do we compute stats for multiple classes?

- We already saw how the “polarity” affects the stats we compute…

Two main approaches. Either:

1. Compute “one-vs-all” 2x2 tables. OR
2. Generalize the 2x2 tables and compute per-class TP / FP / FN based on the diagonals and off-diagonals

1. Compute “one-vs-all” 2x2 tables

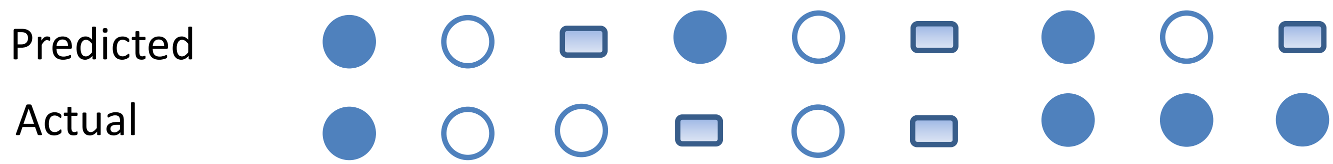


Look for 	Actually Target	Actually Not Target
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

Look for 	Actually Target	Actually Not Target
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

Look for 	Actually Target	Actually Not Target
Selected/Guessed	True Positive (TP)	False Positive (FP)
Not select/not guessed	False Negative (FN)	True Negative (TN)

1. Compute "one-vs-all" 2x2 tables

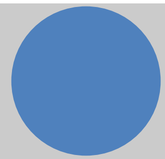
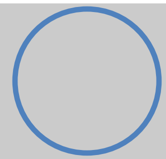
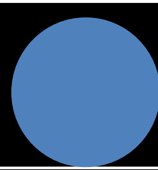
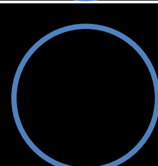


Look for ●	Actually Target	Actually Not Target
Selected/Guessed	2	1
Not select/not guessed	2	4

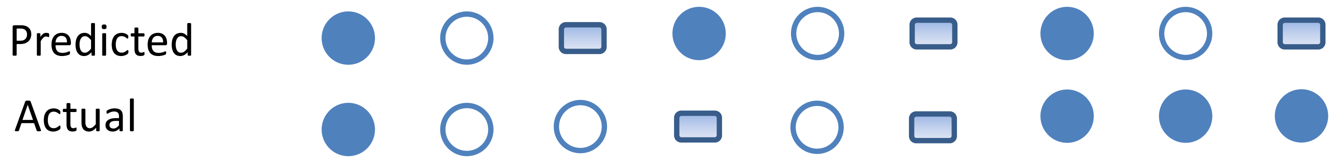
Look for ○	Actually Target	Actually Not Target
Selected/Guessed	2	1
Not select/not guessed	1	5

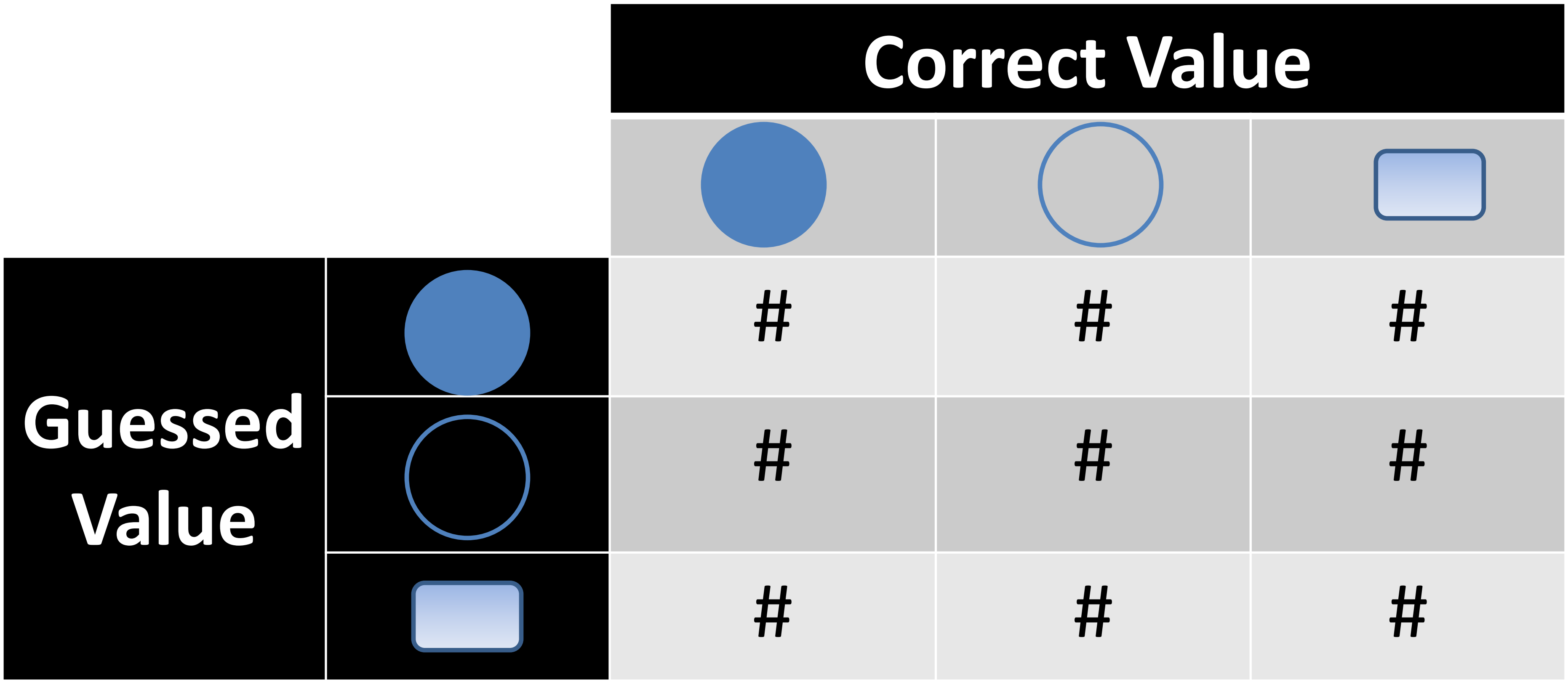
Look for ▭	Actually Target	Actually Not Target
Selected/Guessed	1	2
Not select/not guessed	1	5

2. Generalizing the 2-by-2 contingency table

		Correct Value		
				
Guessed Value		#	#	#
		#	#	#
		#	#	#

2. Generalizing the 2-by-2 contingency table

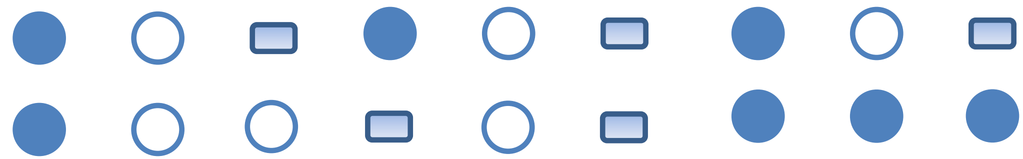


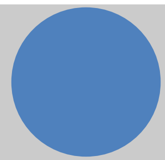
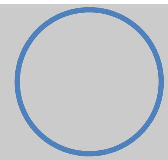
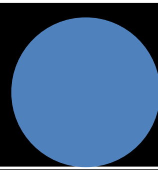
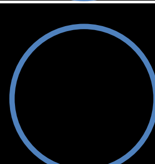


2. Generalizing the 2-by-2 contingency table

Predicted

Actual

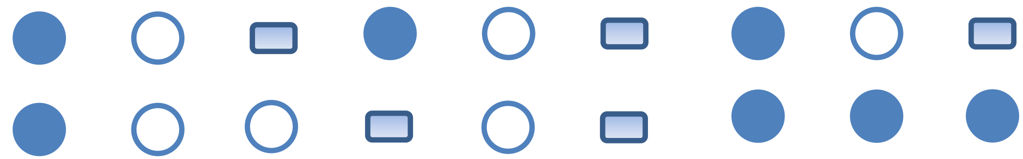


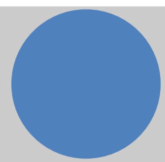
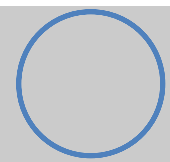
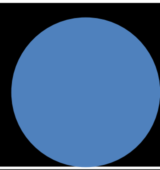
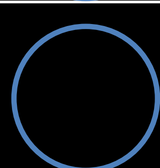
		Correct Value		
				
Guessed Value		2	0	1
		1	2	0
		1	1	1

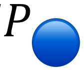
2. Generalizing the 2-by-2 contingency table

Predicted

Actual



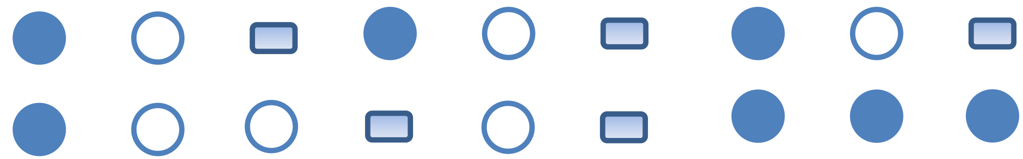
		Correct Value		
				
Guessed Value		2	0	1
		1	2	0
		1	1	1

How do you compute TP?

2. Generalizing the 2-by-2 contingency table

Predicted

Actual



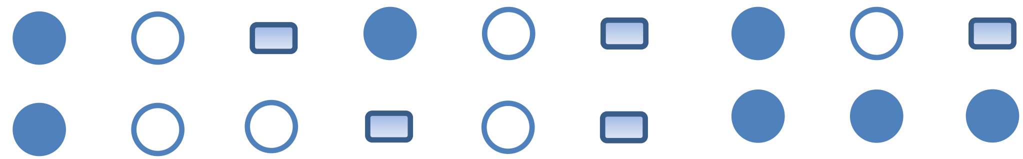
		Correct Value		
		●	○	▭
Guessed Value	●	2	0	1
	○	1	2	0
	▭	1	1	1

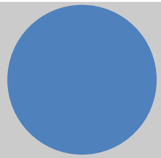
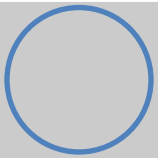
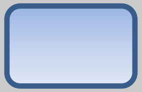
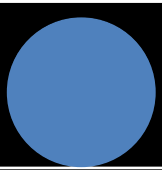
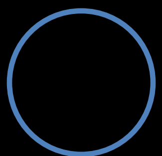
How do you compute $TP_{●}$?

2. Generalizing the 2-by-2 contingency table

Predicted

Actual



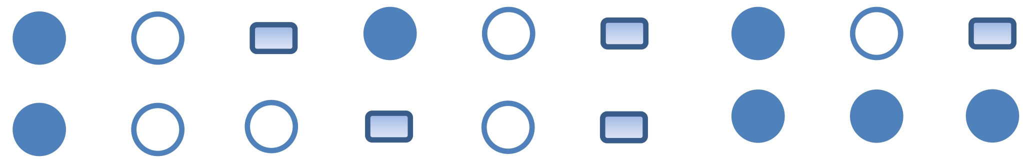
		Correct Value		
				
Guessed Value		2	0	1
		1	2	0
		1	1	1

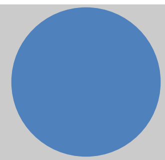
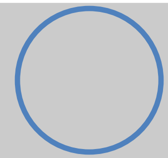
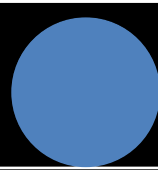
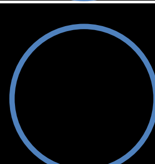
How do you compute FN?

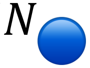
2. Generalizing the 2-by-2 contingency table

Predicted

Actual



		Correct Value		
				
Guessed Value		2	0	1
		1	2	0
		1	1	1

How do you compute FN?

2. Generalizing the 2-by-2 contingency table

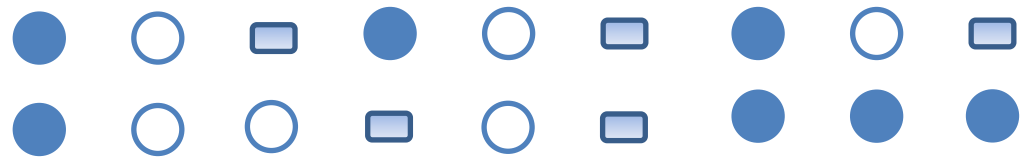
Predicted	●	○	▭	●	○	▭	●	○	▭
Actual	●	○	○	▭	○	▭	●	●	●

		Correct Value		
		●	○	▭
Guessed Value	●	2	0	1
	○	1	2	0
	▭	1	1	1

How do you compute $FP_{\square}$?

2. Generalizing the 2-by-2 contingency table

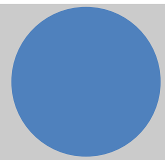
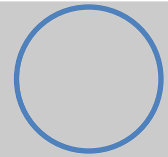
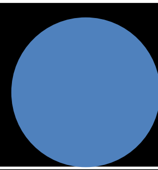
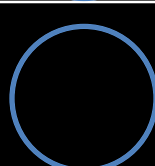
Predicted
Actual



		Correct Value		
		●	○	▭
Guessed Value	●	2	0	1
	○	1	2	0
	▭	1	1	1

How do you compute $FP_{\square}$?

Generalizing the 2-by-2 contingency table

		Correct Value		
				
Guessed Value		#	#	#
		#	#	#
		#	#	#

This is also called a

Confusion Matrix

Generalizing the 2-by-2 contingency table

		Correct Value		
Guessed Value		80	9	11
		7	86	7
		2	8	9

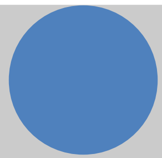
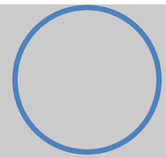
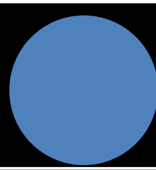
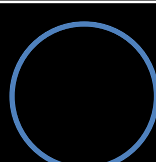
Q: Is this a good result?

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?

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?

Some Classification Metrics

Accuracy

Precision
Recall

AUC (Area Under Curve)

F1

Confusion Matrix

Outline

Classification (Methodology)

Evaluation