

Introduction to Machine Learning: Methodology and Classification Evaluation

Frank Ferraro – ferraro@umbc.edu

CMSC 473/673

Outline

Classification (Methodology)

Evaluation

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
...	

Text Classification: Supervised Machine Learning

Assigning subject
categories, topics, or
genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

...

Input:

a document d

a fixed set of classes $C = \{c_1, c_2, \dots, c_J\}$

A training set of m hand-labeled
documents $(d_1, c_1), \dots, (d_m, c_m)$

Output:

a learned classifier γ that maps
documents to classes

Naïve Bayes

Logistic regression

Support-vector
machines

k-Nearest Neighbors

...

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification			
Multi-class Classification			
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	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			
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	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			
Multi-task Classification			

Classification Types (Terminology)

Name	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			

Classification Types (Terminology)

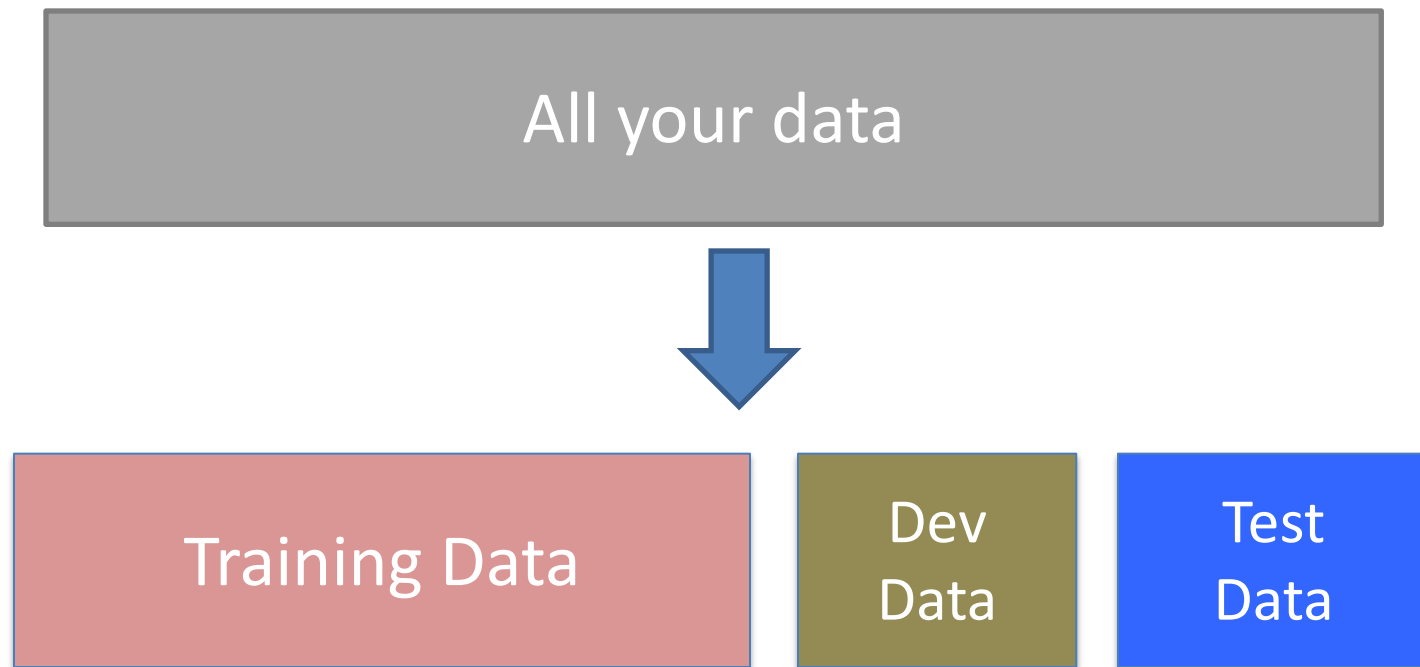
Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, ...}
Multi-label Classification	1	> 2	Sentiment: Choose multiple 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

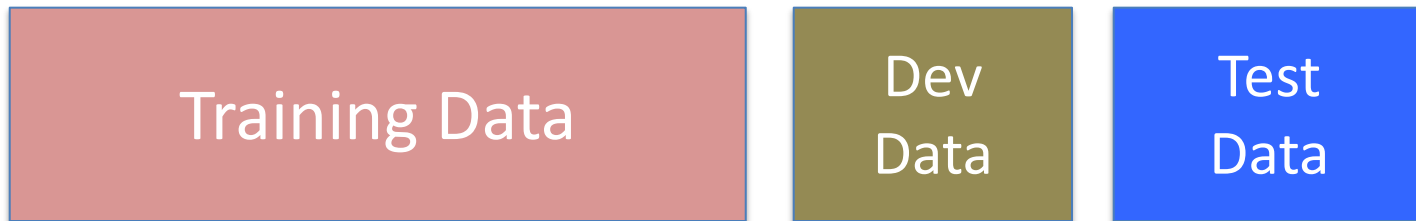


DON'T ITERATE ON YOUR TEST DATA

Experimenting with Machine Learning Models

What is “correct?”

What is working “well?”

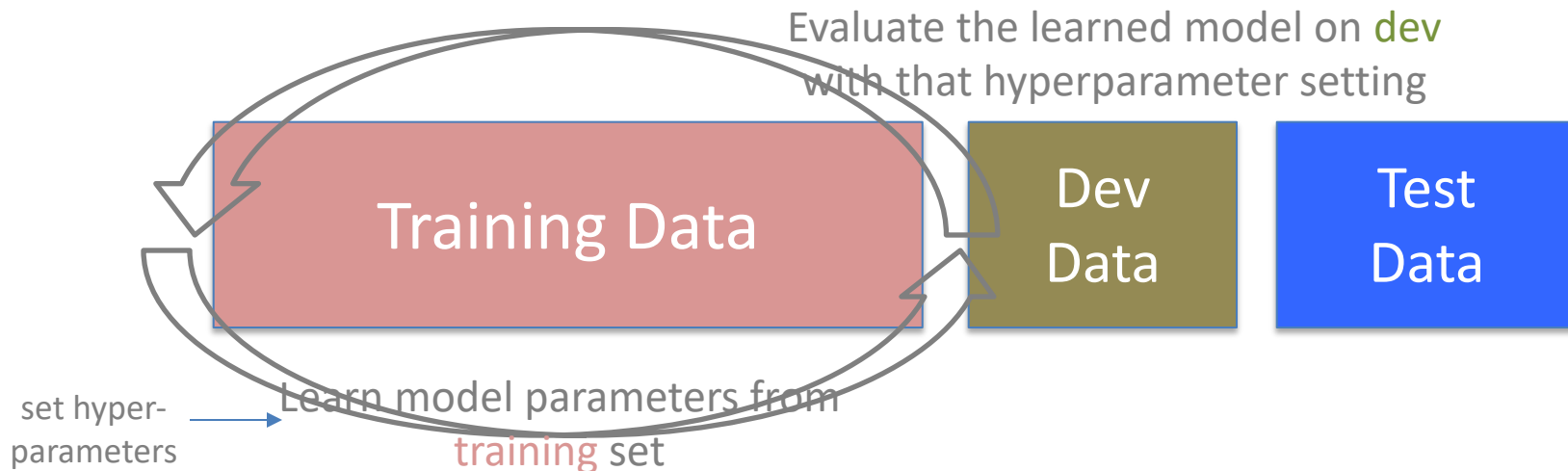


set hyper-parameters → Learn model parameters from
training set

Experimenting with Machine Learning Models

What is “correct?”

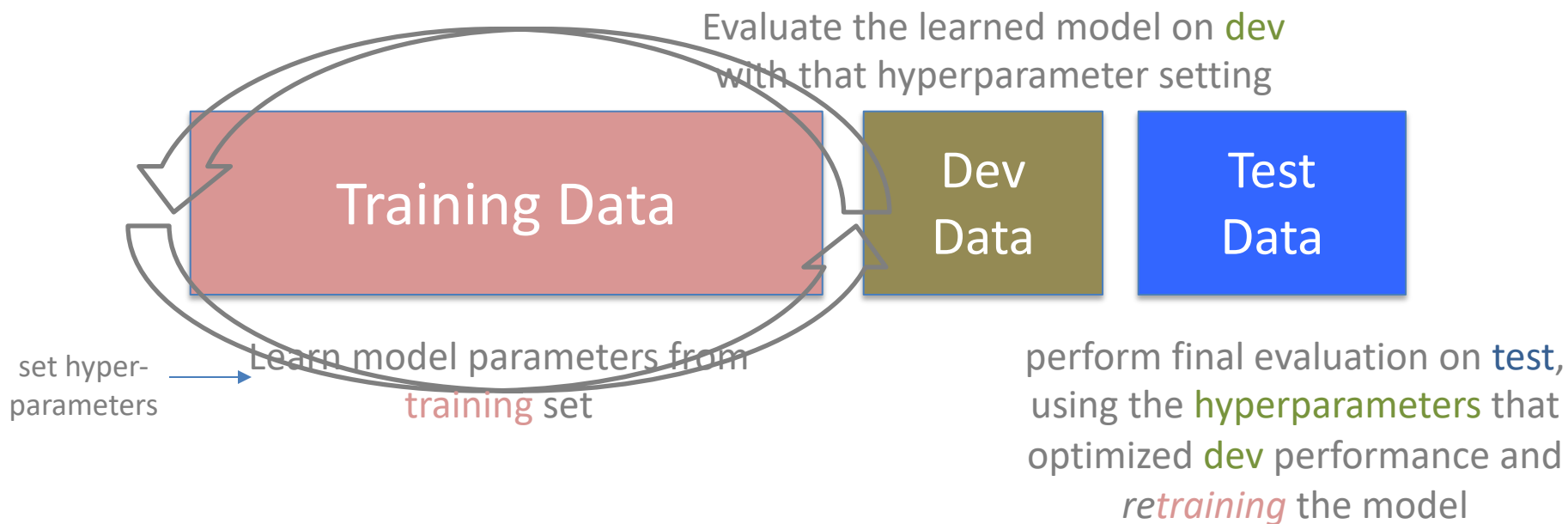
What is working “well?”



Experimenting with Machine Learning Models

What is “correct?”

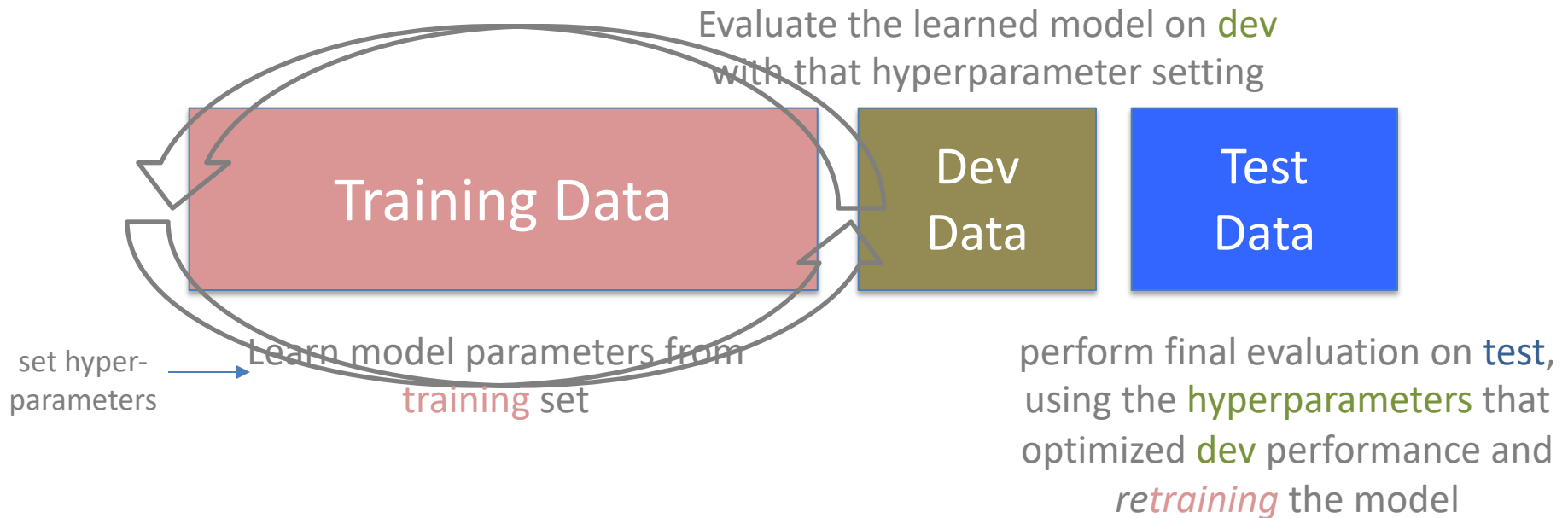
What is working “well?”



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.)

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
- ...

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

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

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

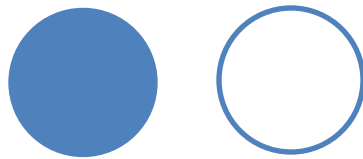
AUC (Area Under Curve)

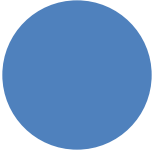
F1

Confusion Matrix

Classification Evaluation: the 2-by-2 contingency table

Let's assume there are two classes/labels



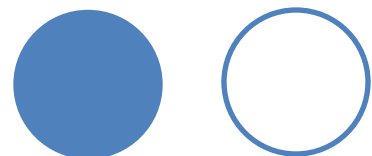
Assume  is the “positive” label

Given X , our classifier predicts either label

$$p(\text{solid blue circle} | X) \text{ vs. } p(\text{empty blue circle} | X)$$

Classification Evaluation: the 2-by-2 contingency table

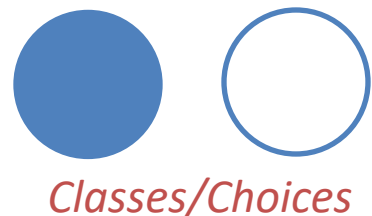
<i>What label does our system predict? (↓)</i>	<i>What is the actual label?</i>	
	Actually Correct	Actually Incorrect
Selected/ Guessed		
Not selected/ not guessed		



Classes/Choices

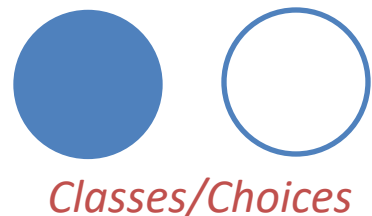
Classification Evaluation: the 2-by-2 contingency table

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



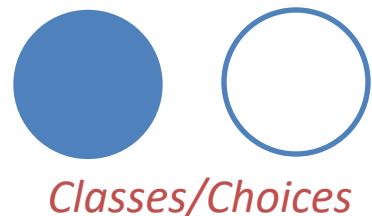
Classification Evaluation: the 2-by-2 contingency table

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



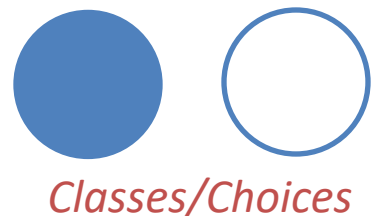
Classification Evaluation: the 2-by-2 contingency table

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



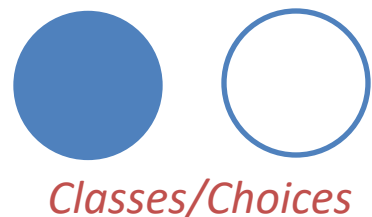
Classification Evaluation: the 2-by-2 contingency table

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



Classification Evaluation: the 2-by-2 contingency table

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



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

Contingency Table Example

Predicted:					
Actual:					

Contingency Table Example

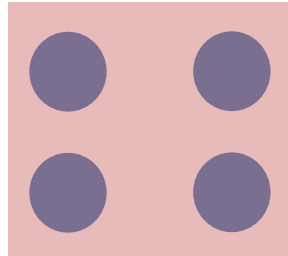
Predicted:     

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)

Contingency Table Example

Predicted:



Actual:



What is the actual label?		
What label does our system predict? (↓)	Actually Correct	Actually Incorrect
	True Positive (TP) = 2	False Positive (FP)
	False Negative (FN)	True Negative (TN)

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) = 2

False Positive
(FP) = 1

**Not selected/
not guessed**

False Negative
(FN)

True Negative
(TN)

Contingency Table Example

Predicted:



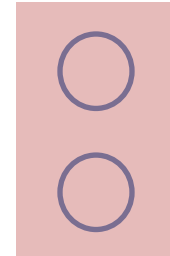
Actual:



	What is the actual label?	
What label does our system predict? (↓)	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)

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) = 2	False Positive (FP) = 1
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? (↓)	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

Classification Evaluation:

Accuracy, Precision, and Recall

Accuracy: % of items correct

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

	Actually Correct	Actually Incorrect
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

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

Precision: % of selected items that are correct

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

	Actually Correct	Actually Incorrect
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

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

Precision: % of selected items that are correct

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

Recall: % of correct items that are selected

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

	Actually Correct	Actually Incorrect
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

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

Precision: % of selected items that are correct

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

Min: 0 😞

Max: 1 😊

Recall: % of correct items that are selected

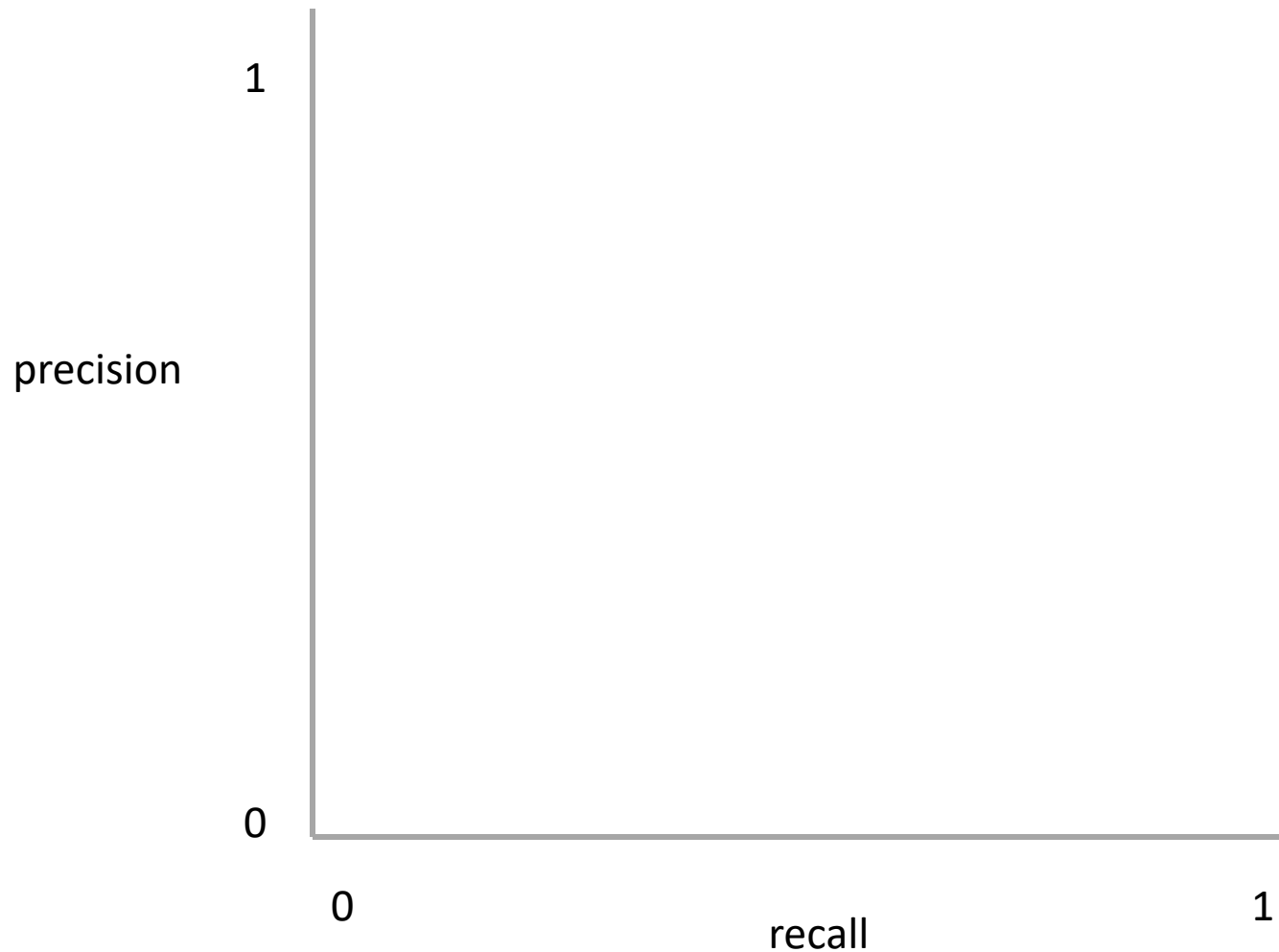
$$\frac{TP}{TP + FN}$$

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

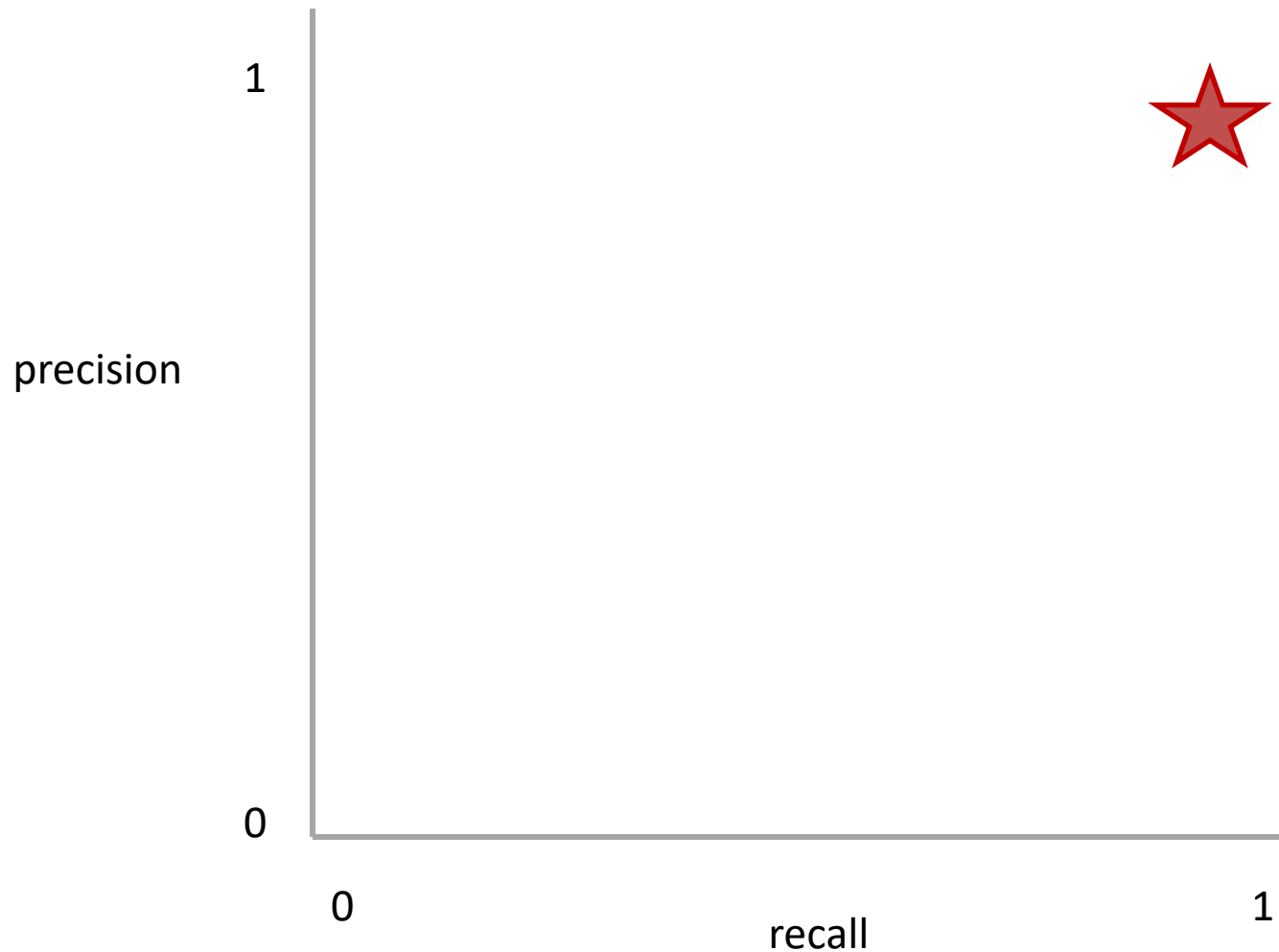
Precision and Recall Present a Tradeoff

Q: Where do you
want your ideal

model ?



Precision and Recall Present a Tradeoff

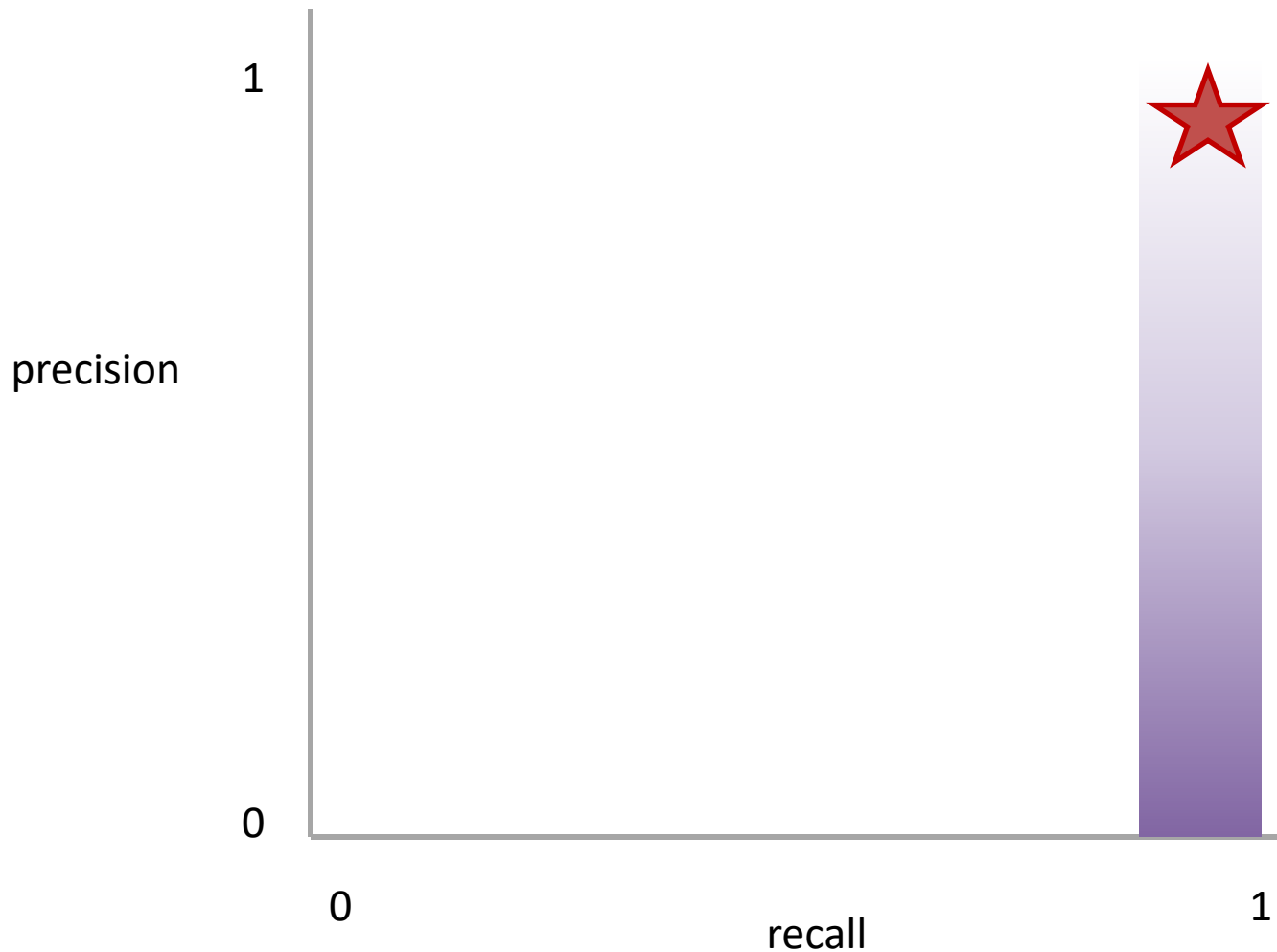


Q: Where do you want your ideal

model ?

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

Precision and Recall Present a Tradeoff

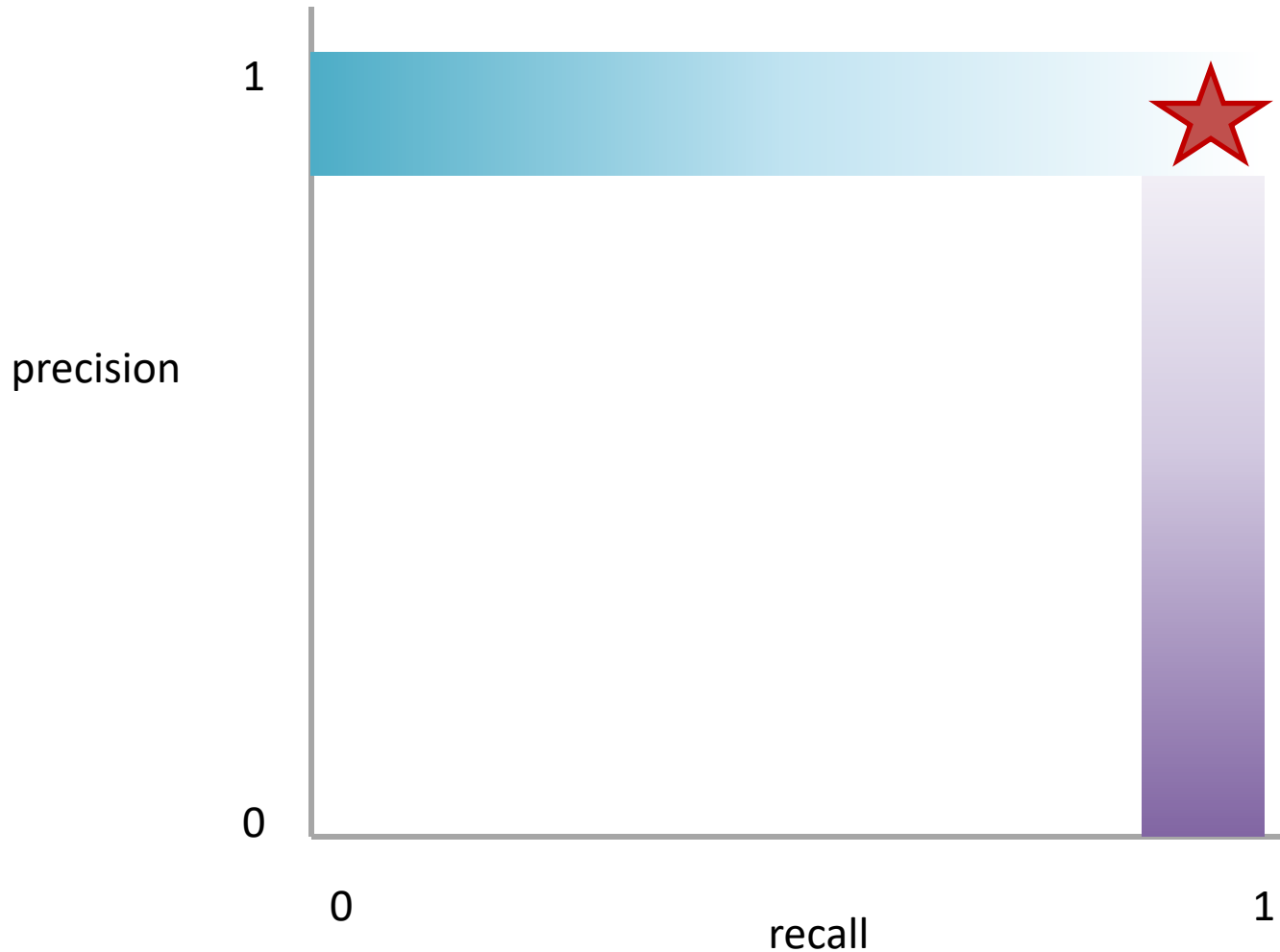


Q: Where do you want your ideal **model** ?

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



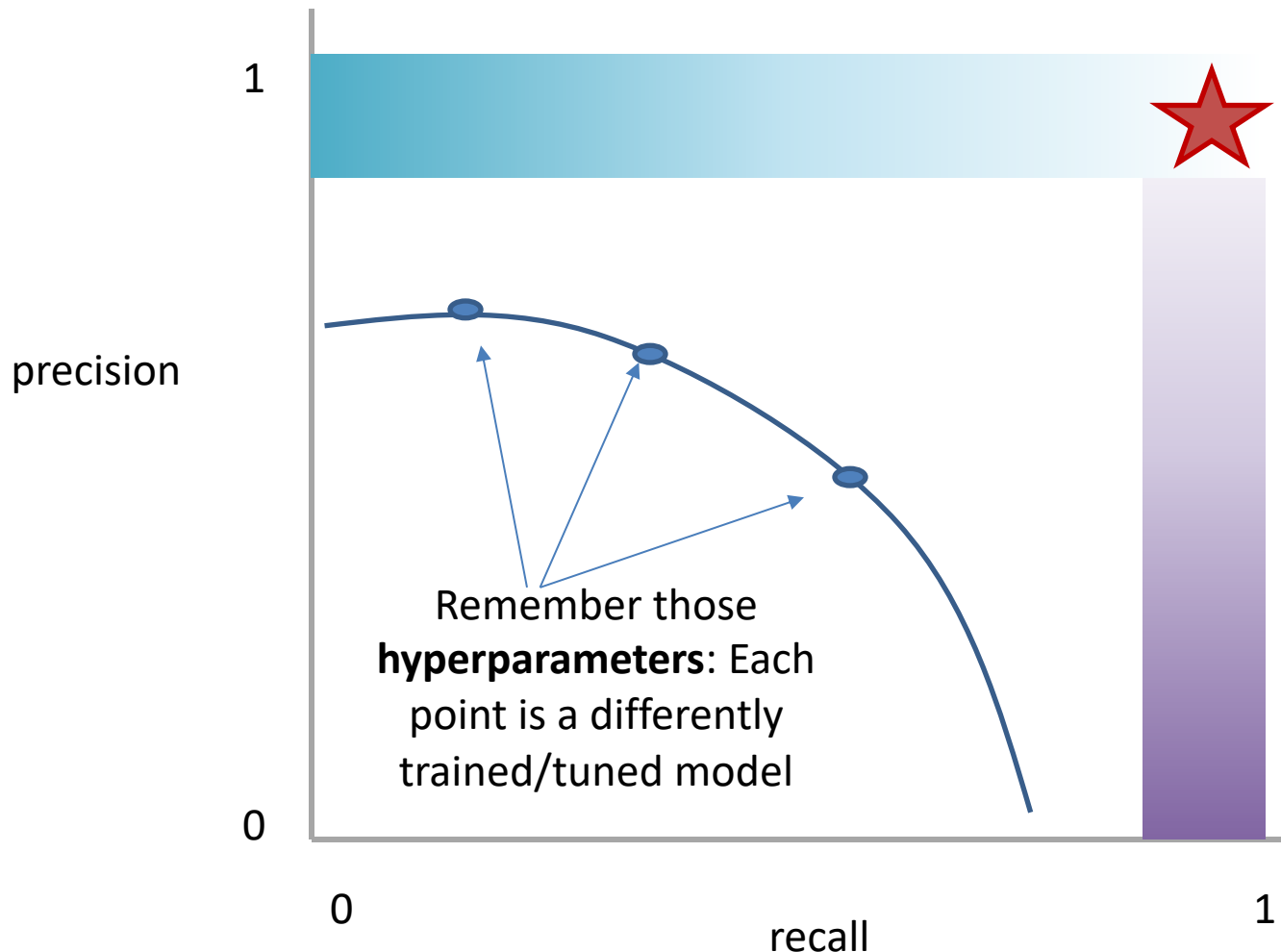
Q: Where do you want your ideal

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



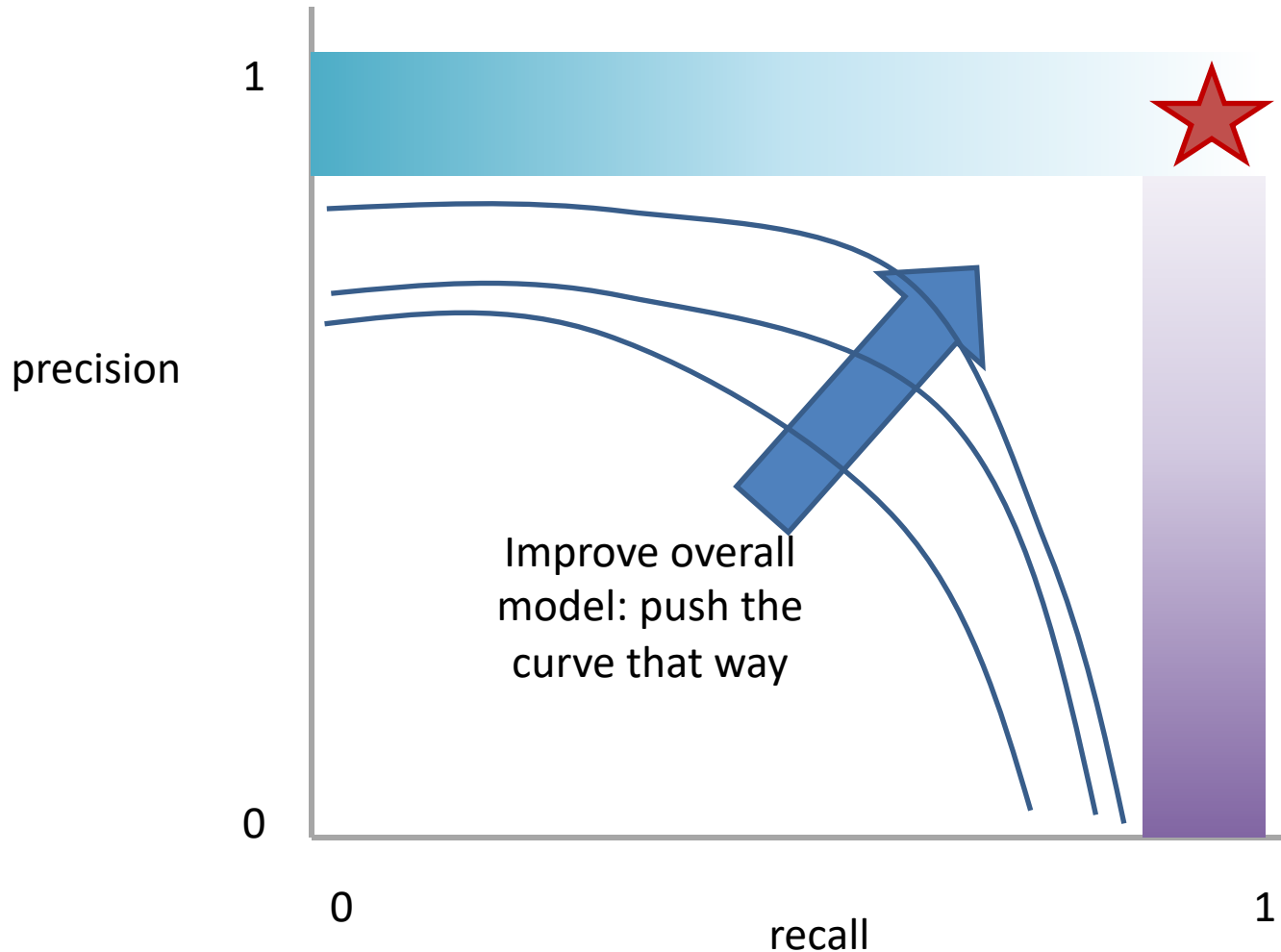
Q: Where do you want your ideal **model** ?

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 ?

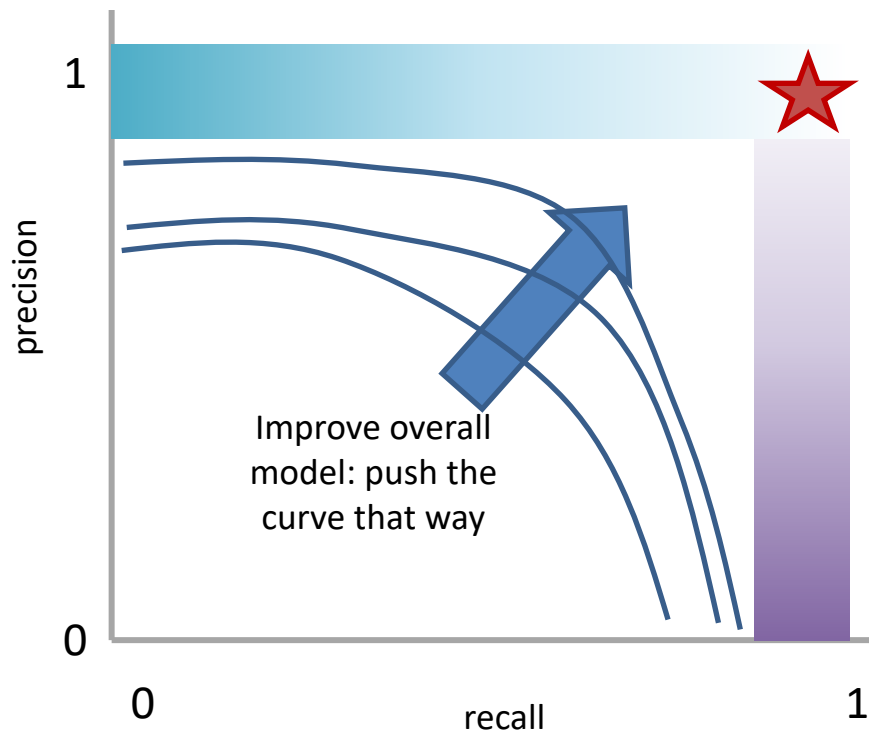
Q: You have a model that always identifies correct instances. Where on this graph is it?

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

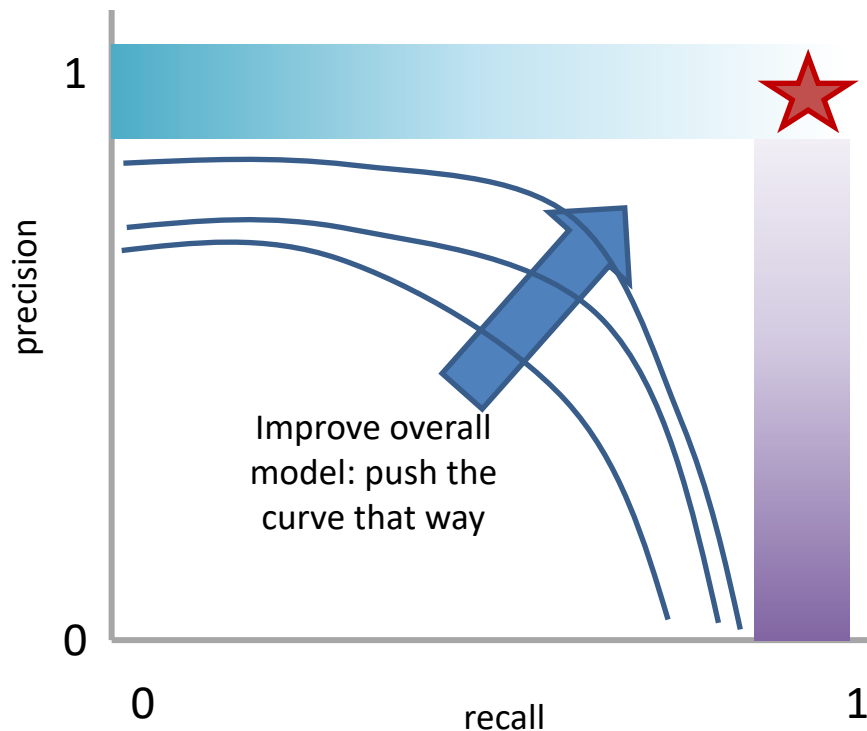


Min AUC: 0 😞

Max AUC: 1 😊

Measure this Tradeoff: Area Under the Curve (AUC)

AUC measures the area under this tradeoff curve



Min AUC: 0 😞

Max AUC: 1 😊

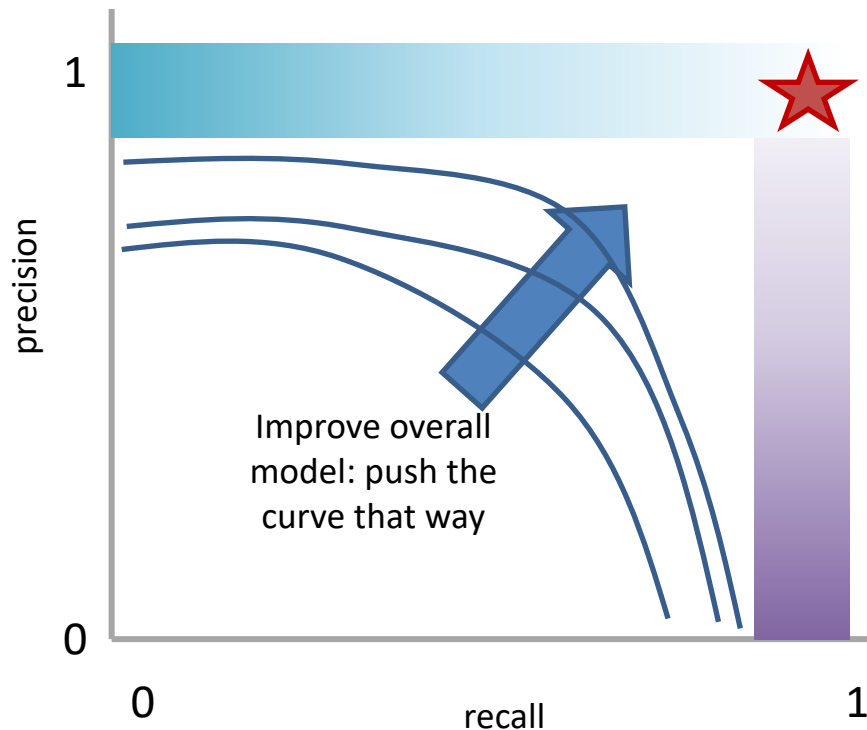
1. Computing the curve

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)

AUC measures the area under this tradeoff curve



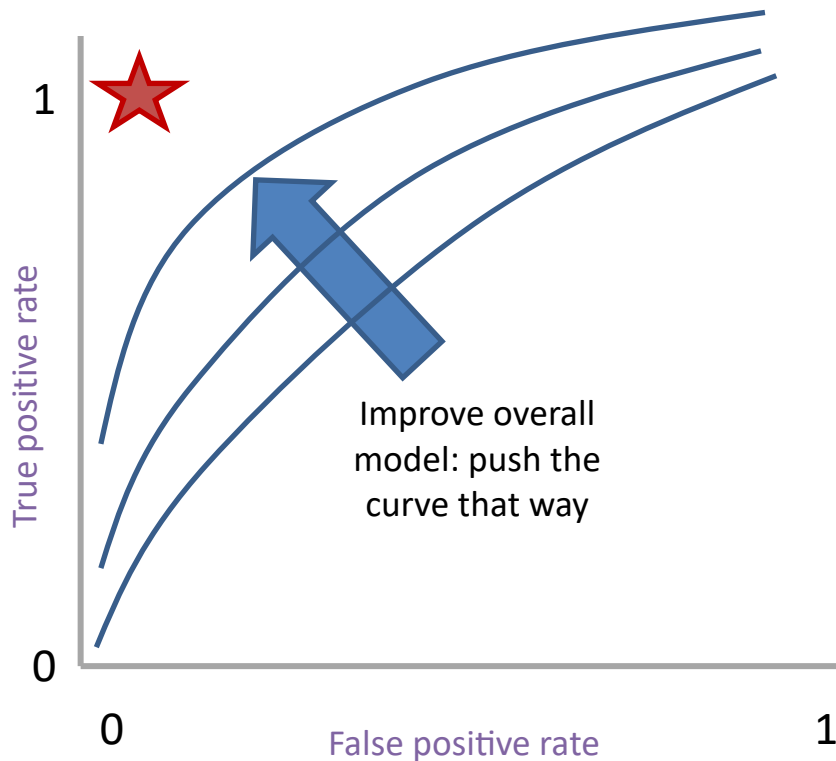
Min AUC: 0 😞

Max AUC: 1 😊

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

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

Balanced F1 measure: $\beta=1$

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

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.

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

Macroaveraging: Compute performance for each class, then average.

$$\text{macroprecision} = \frac{1}{C} \sum_c \frac{TP_c}{TP_c + FP_c} = \frac{1}{C} \sum_c \text{precision}_c$$

$$\text{macrorecall} = \frac{1}{C} \sum_c \frac{TP_c}{TP_c + FN_c} = \frac{1}{C} \sum_c \text{recall}_c$$

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

$$\text{microprecision} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c}$$

$$\text{microrecall} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FN_c}$$

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

Macroaveraging: Compute performance for each class, then average.

when to prefer the macroaverage?

$$\text{macroprecision} = \frac{1}{C} \sum_c \frac{TP_c}{TP_c + FP_c} = \frac{1}{C} \sum_c \text{precision}_c$$

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

when to prefer the microaverage?

$$\text{microprecision} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c}$$

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

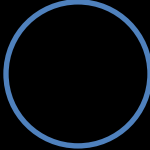
	Truth : yes	Truth : no
Classifier: yes	100	20
Classifier: no	20	1860

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

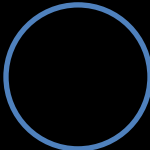
Microaveraged precision: $100/120 = .83$

Microaveraged score is dominated by score on frequent classes

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

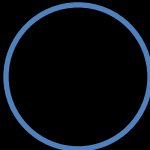
		Correct Value		
				
Guessed Value		#	#	#
		#	#	#
		#	#	#

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

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

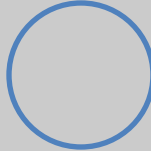
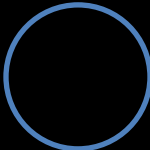
Q: Is this a good result?

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?

Some Classification Metrics

Accuracy

Precision

Recall

AUC (Area Under Curve)

F1

Confusion Matrix

Outline

Classification (Methodology)

Evaluation