

# ML Evaluation

---

CMSC 473/673 - NATURAL LANGUAGE PROCESSING

*Slides modified from Dr. Frank Ferraro & Cynthia Matuszek*

# Learning Objectives

---

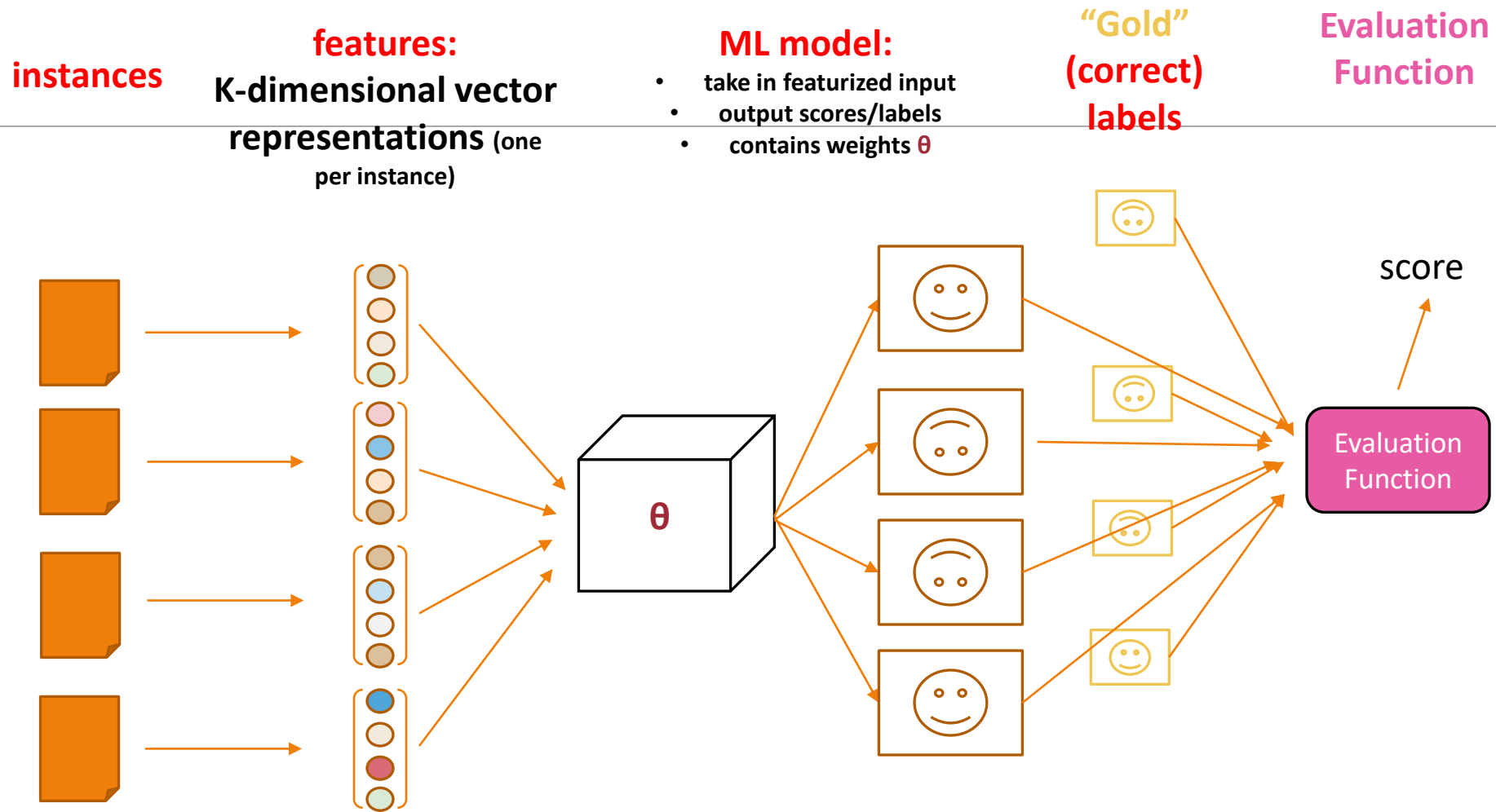
Distinguish between types of ML problems and models

Fill out a contingency table

Calculate accuracy, precision, and recall

Develop an intuition about precision & recall

# Review: ML/NLP Framework for Prediction



# Review: Classification Types (Terminology)

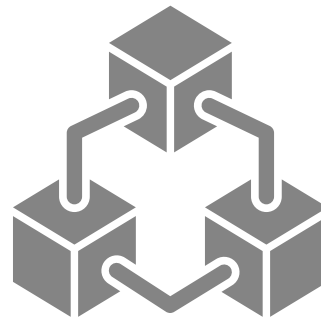
Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	$> 2$	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, ...}
Multi-label Classification	1	$> 2$	Sentiment: Choose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

# How do we learn models?

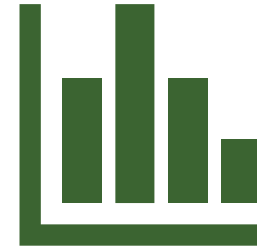
---



Take past experiences  
(lots of data; corpus)



Find patterns  
(the ML algorithm)

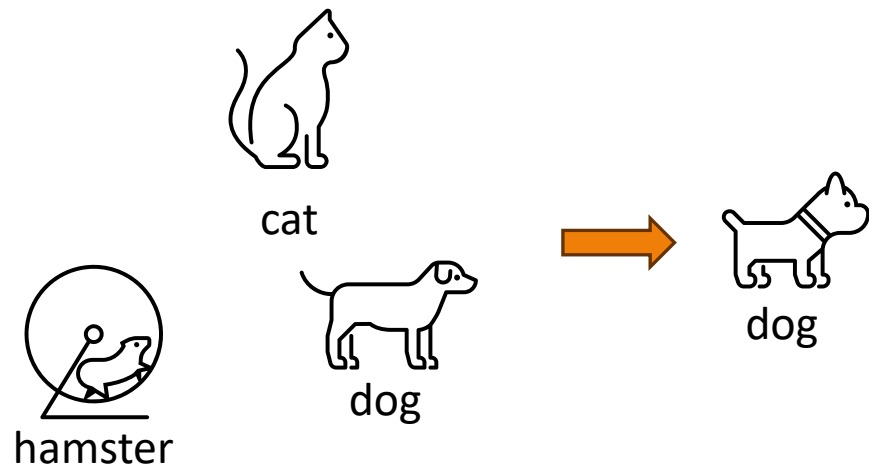


Use on new experiences  
(save & test the model)

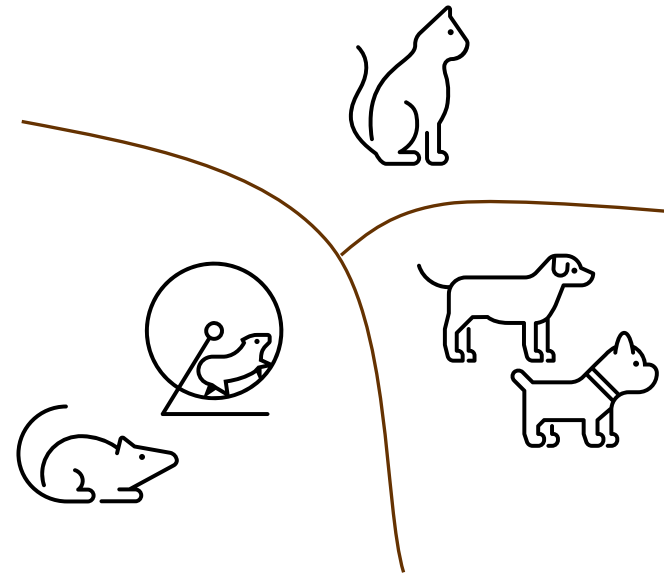
# Types of Learning

---

## SUPERVISED LEARNING



## UNSUPERVISED LEARNING



# Types of Learning

---

## SUPERVISED LEARNING

Data has feedback (labels)

Data consists of input-output pairs

Learn mapping from input to output

*Examples:*

- Dataset classification
- How likely is it that this person will get into a car accident?

## UNSUPERVISED LEARNING

No explicit feedback in data

Learn patterns directly from data

*Examples:*

- Clustering
- Do these people fall under multiple groups?

# What are some other examples of these?

---

## **SUPERVISED LEARNING**

- Machine translation
- Object segmentation (vision)
- Document classification

## **UNSUPERVISED LEARNING**

- Clustering
- Language modeling



# The Machine Learning Framework

---

$$y = f(x)$$

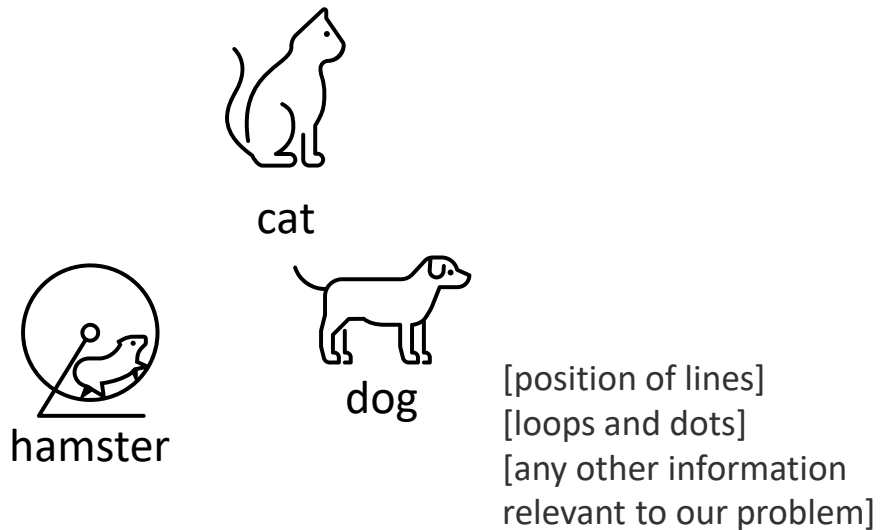
output      prediction function      feature(s) of input

**Training:** given a *training set* of labeled examples  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ , estimate the prediction function  $f$  by minimizing the prediction error on the training set

**Testing:** apply  $f$  to a never before seen *test example*  $\mathbf{x}$  and output the predicted value  $y = f(\mathbf{x})$

# How do we learn models?

---



Have data with  
features extracted  
(and possibly labels)

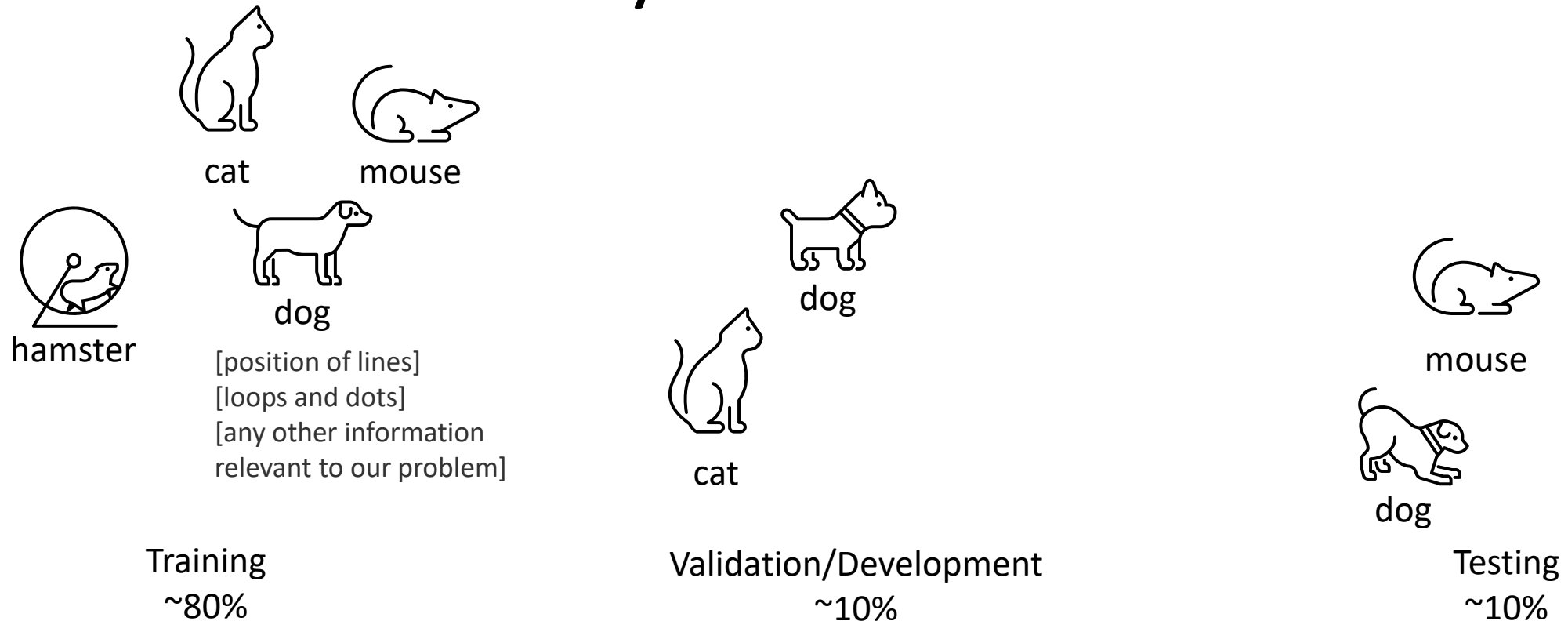
$P(\text{hamster} \mid [\text{line in this position}], \dots)$

$P(\text{dog} \mid [\text{line in this other position}], \dots)$

Learn associations  
between features  
and labels

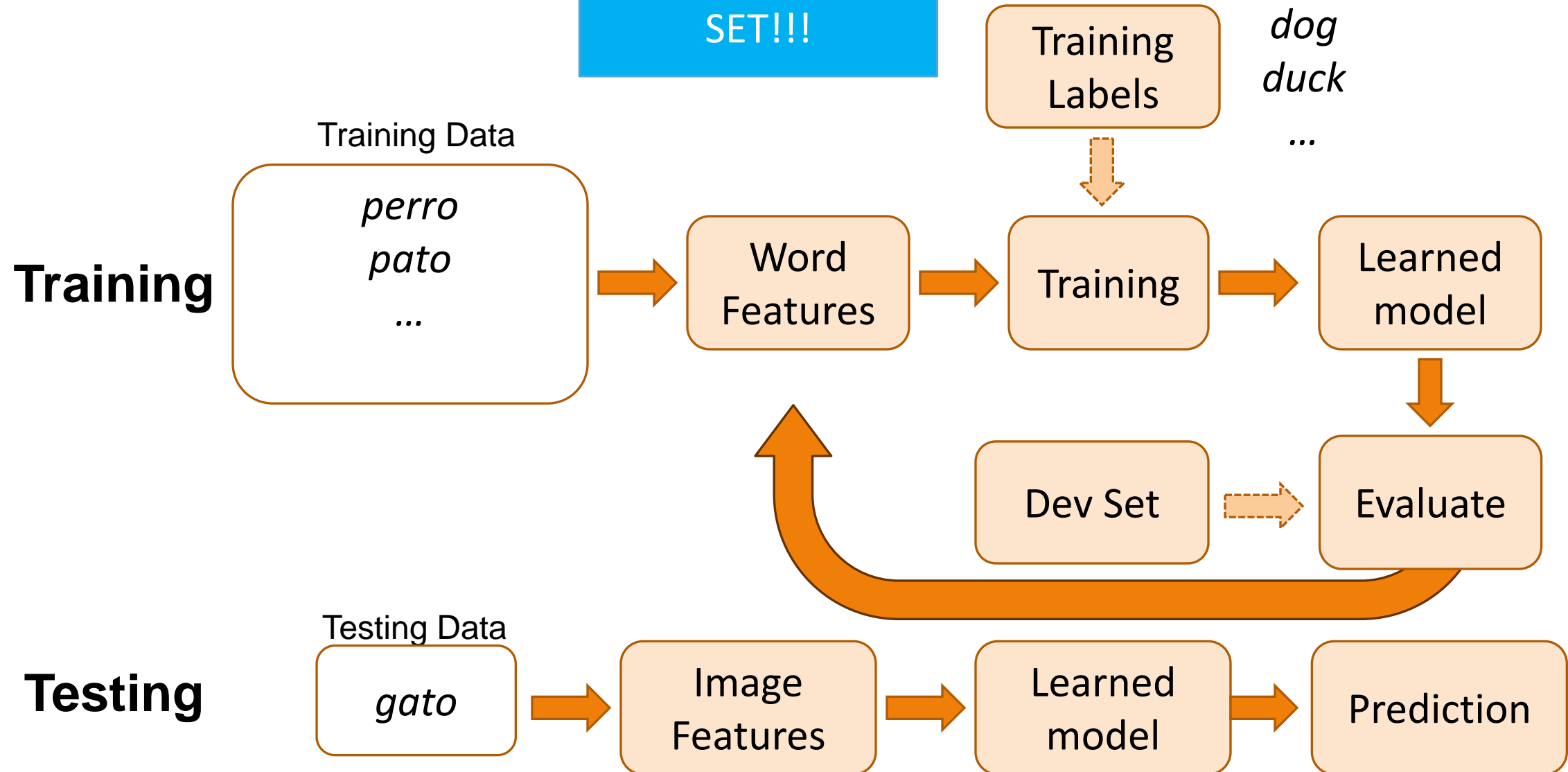
# Dividing up data for Training

## Why would we do this?



# Steps

DO NOT ITERATE  
ON THE TESTING  
SET!!!



# Types of models

---

## CLASSIFICATION

Model outputs comes from a finite set of values

Discrete result

*Examples:*

- What type of animal is this a picture of?
- Predicting the weather (sunny, cloudy, or rainy?)
- Ranking: Is this result *better* than this result?

## REGRESSION

Model outputs are continuous values

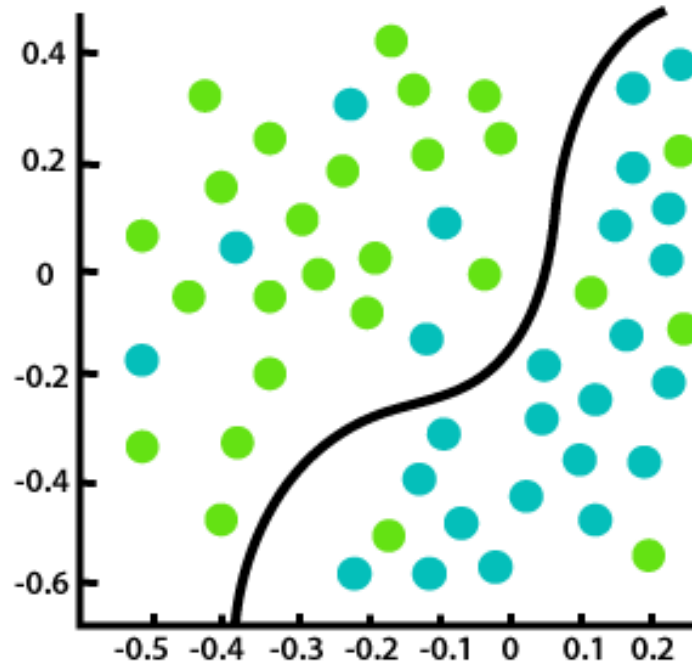
Continuous result

*Examples:*

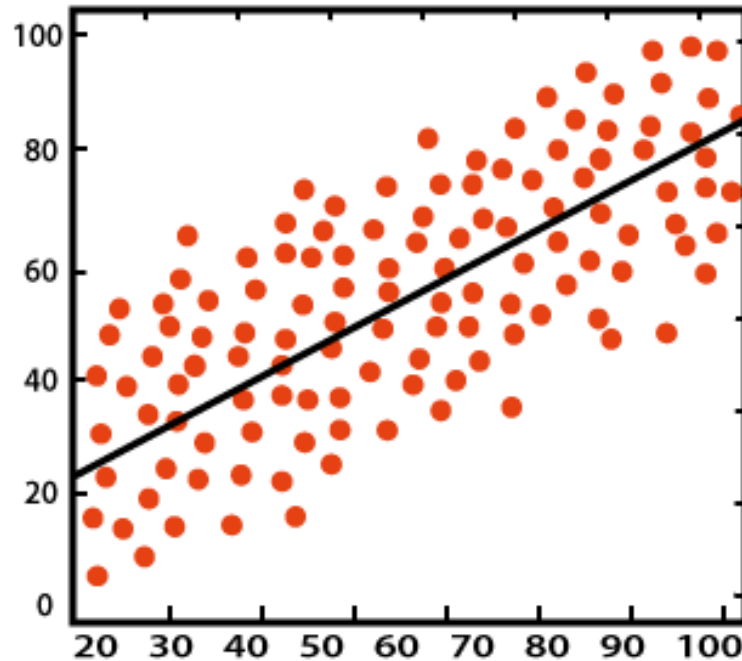
- How far will I move if I drive my motors at this speed for 1 second?
- Predicting the weather (temperature)
- Ranking: *how good* is this result?

# Types of models

---



Classification



Regression

# What are some other examples of these?

---

## CLASSIFICATION

Tone tagging

Sentiment classification

Named entity recognition

## REGRESSION

Quantity/scale of how much it sounds like a specific author

Numerical sentiment value

Political “score” from document

Likelihoods

Predicted Goodreads score

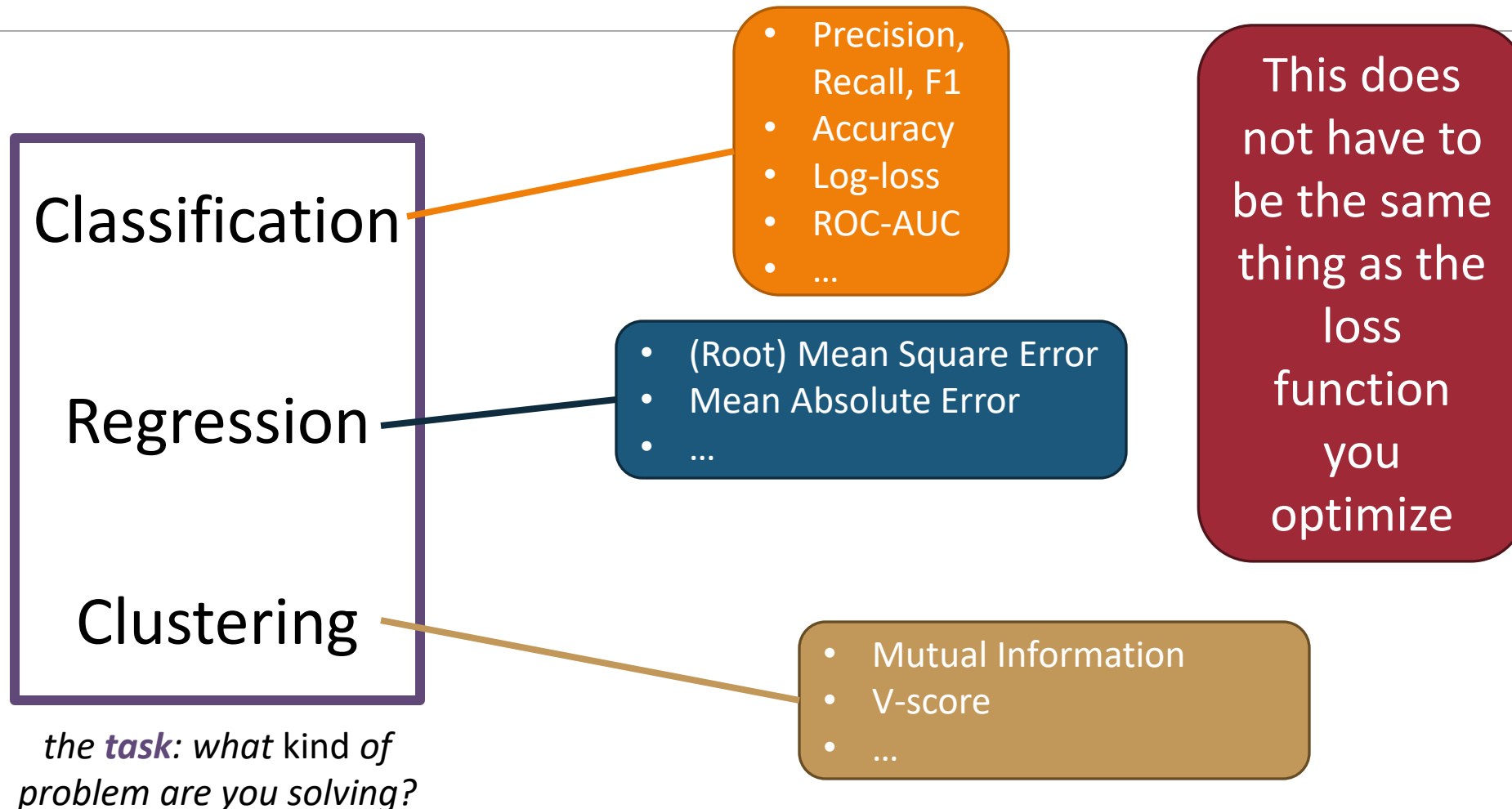
# Types of Algorithms

---

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	classification or categorization	clustering
<i>Continuous</i>	regression	dimensionality reduction



# Central Question: How Well Are We Doing?



# 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

# Implementation: How To

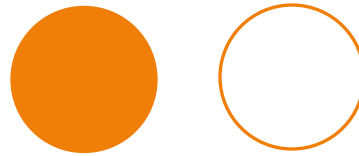
---

1. scikit-learn: [sklearn.metrics](#)
  - very stable
2. huggingface [evaluate](#) module
  - community input
  - sometimes are based on sklearn
3. implement your own

# Classification Evaluation: the 2-by-2 contingency table


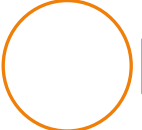
---

Assumption 1: There are two classes/labels



Assumption 2:  is the “positive” label

Assumption 3: Given  $X$ , our classifier produces a score for each possible label

$$p(\text{  | X) \text{ vs. } p(\text{  | X)$$

# Examining Assumption 3

---

Given  $X$ , our classifier produces a score for each possible label

$$p(\text{●} | X) \text{ vs. } p(\text{○} | X)$$

Normally (\*but this can be adjusted!)

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

# Example of `argmax`

---

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
...	

# Example of `argmax`

---

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
<b>TERRORISM</b>	<b>.48</b>
SPORTS	.0001
TECH	.39
HEALTH	.0001
FINANCE	.0002
...	







# Classification Evaluation: the 2-by-2 contingency table

<i>What label does our system predict? (↓)</i>	<i>What is the actual label?</i>	
	<b>Actual Target Class ("●")</b>	<b>Not Target Class ("○")</b>
<b>Selected/ Guessed ("●")</b>		
<b>Not selected/ not guessed ("○")</b>		







# Classification Evaluation: the 2-by-2 contingency table

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









# Classification Evaluation: the 2-by-2 contingency table

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









# Classification Evaluation: the 2-by-2 contingency table

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

# Classification Evaluation: the 2-by-2 contingency table

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

# Classification Evaluation: the 2-by-2 contingency table













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



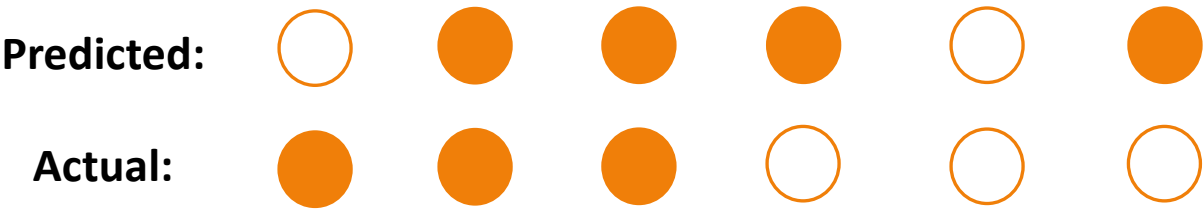
Construct this table by *counting*  
the number of TPs, FPs, FNs, TNs

# Contingency Table Example

---

<b>Predicted:</b>						
<b>Actual:</b>						

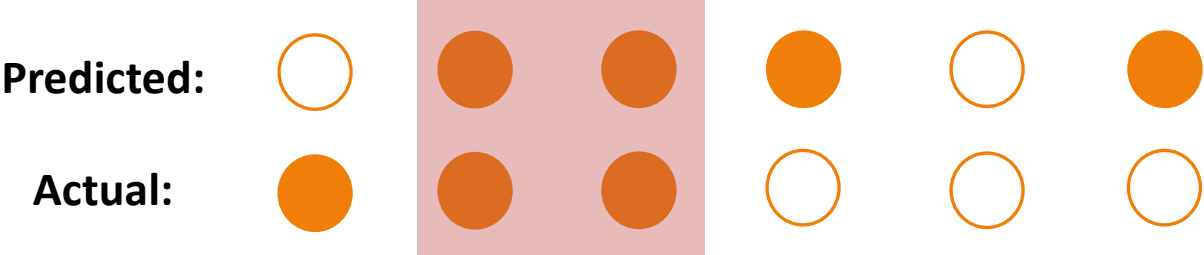
# Contingency Table Example



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

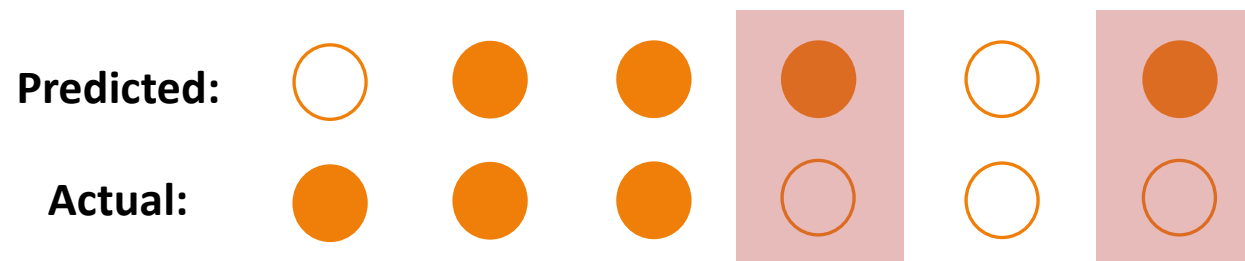


# Contingency Table Example



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



<i>What is the actual label?</i>		
<i>What label does our system predict? (↓)</i>	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:

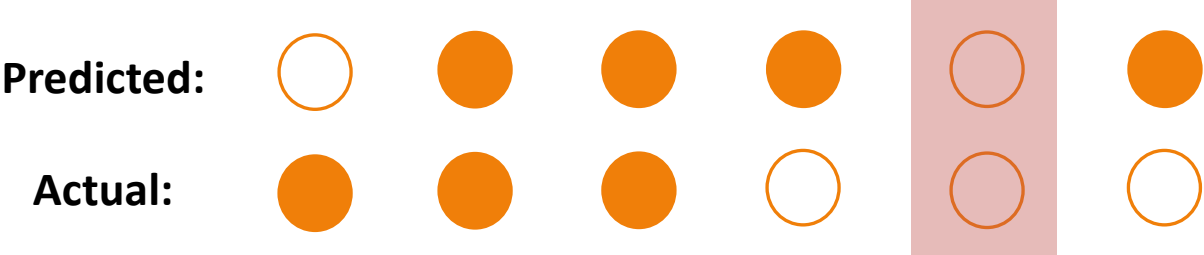


Actual:









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



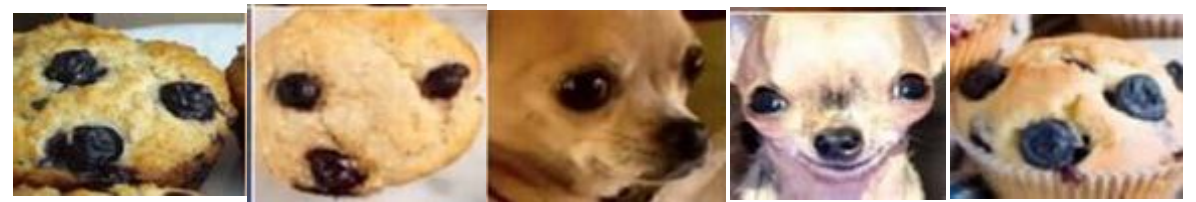
	What is the actual label?	
What label does our system predict? (↓)	Actual Target Class ("●")	Not Target Class ("○")
Selected/ Guessed ("●")	True Positive (TP) = 2	False Positive (FP) = 2
Not selected/ not guessed ("○")	False Negative (FN) = 1	True Negative (TN) = 1

# 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



# Knowledge Check

Fill out the contingency table for this example.  
Your target class is Dog.

**Actual:**

Blueberry Blueberry Dog Dog Blueberry

**Predicted:**

Blueberry Dog Dog Blueberry Blueberry

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

<https://petcentral.chewy.com/are-blueberries-safe-for-dogs-and-everything-else-you-could-possibly-want-to-know-about-dogs-and-blueberries/>

# Classification Evaluation: Accuracy, Precision, and Recall

**Accuracy:** % of items correct

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

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

**Precision:** % of selected items that are correct

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

	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

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

**Precision:** % of selected items that are correct

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

**Recall:** % of correct items that are selected

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

	Actually 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

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

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


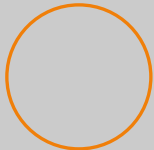

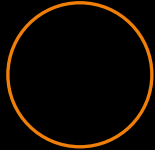
# The Importance of “Polarity” in Binary Classification

---

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


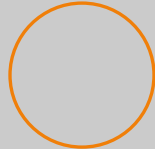



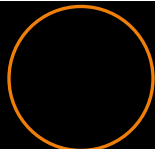


Are you trying to find  or ?

# 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		<i>TP</i> 	<i>FP</i> 
		<i>FN</i> 	<i>TN</i> 

# The Importance of “Polarity” in Binary Classification

**Predicted:** ○ ● ● ● ○ ●

---


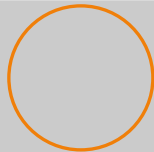

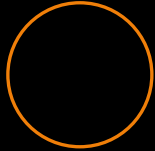
**Actual:** ● ● ● ○ ○ ○

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

What are the  
accuracy, recall, and  
precision values?


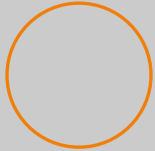



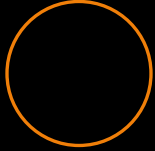


Accuracy: 50%  
Recall: 66.67%  
Precision: 50%

# 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


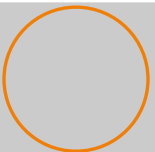



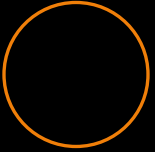


		Correct Value	
			
Guessed Value		$TN$ 	$FN$ 
		$FP$ 	$TP$ 



# The Importance of “Polarity” in Binary Classification

Predicted:      

Actual:      




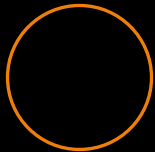
		Correct Value	
			
Guessed Value		$TN$  = 2	$FN$  = 2
		$FP$  = 1	$TP$  = 1

What are the  
accuracy, recall, and  
precision values?

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

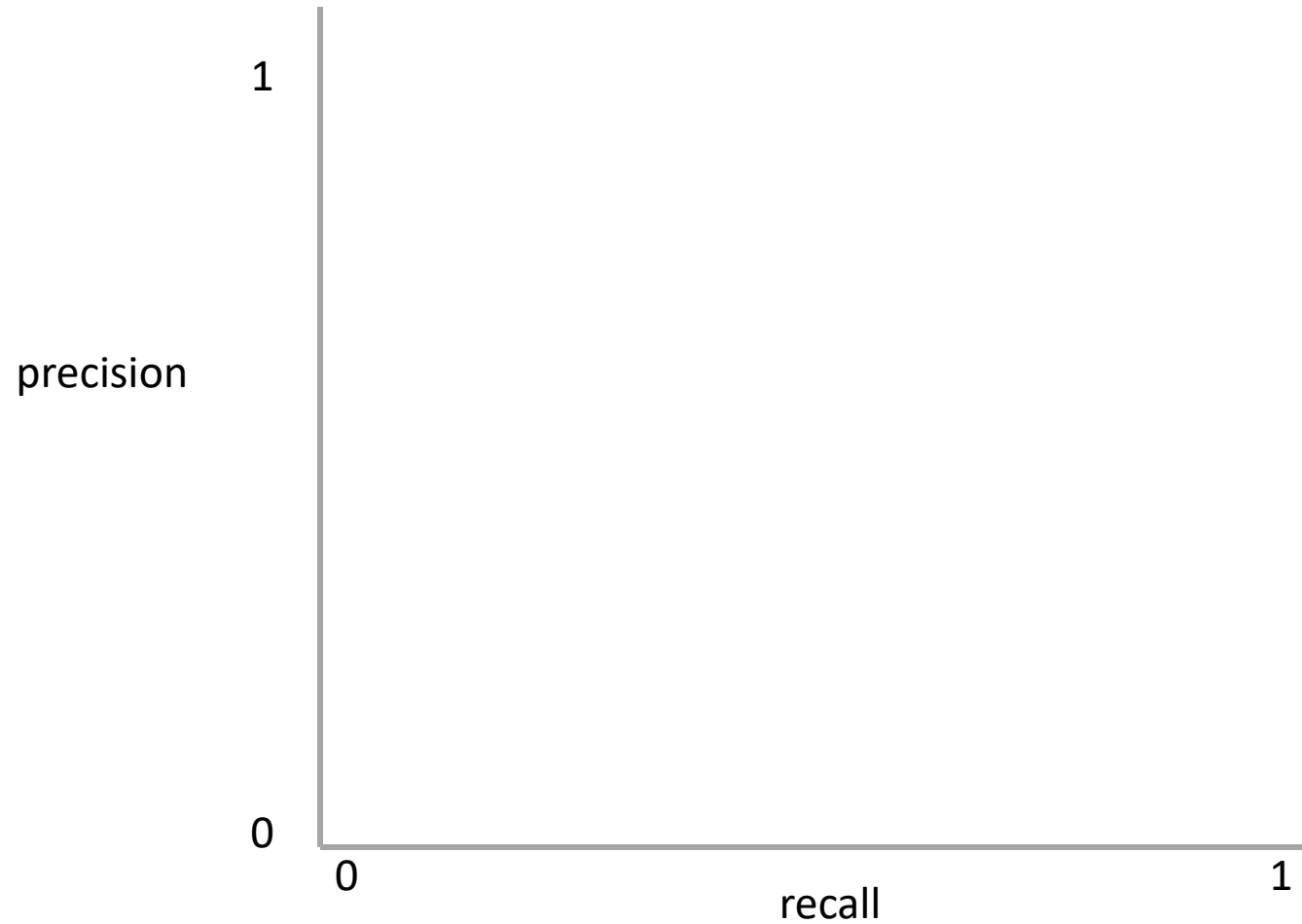
# The Importance of “Polarity” in Binary Classification

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

		Correct Value	
			
Guessed Value		$TP \text{ }  = TN \text{ } $	$FP \text{ }  = FN \text{ } $
		$FN \text{ }  = FP \text{ } $	$TN \text{ }  = TP \text{ } $

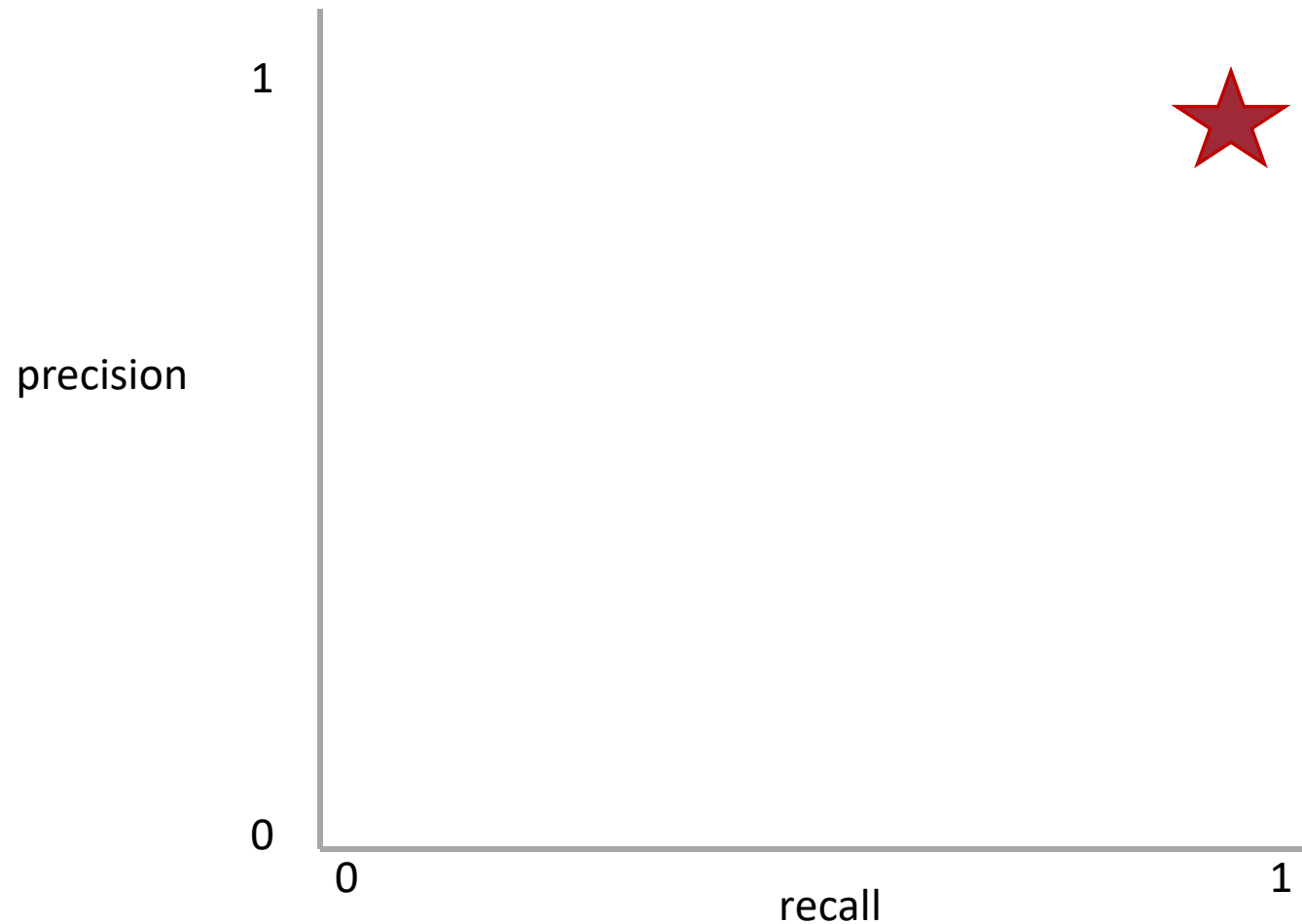
# Precision and Recall Present a Tradeoff

Q: Where do you  
want your ideal  
model ?

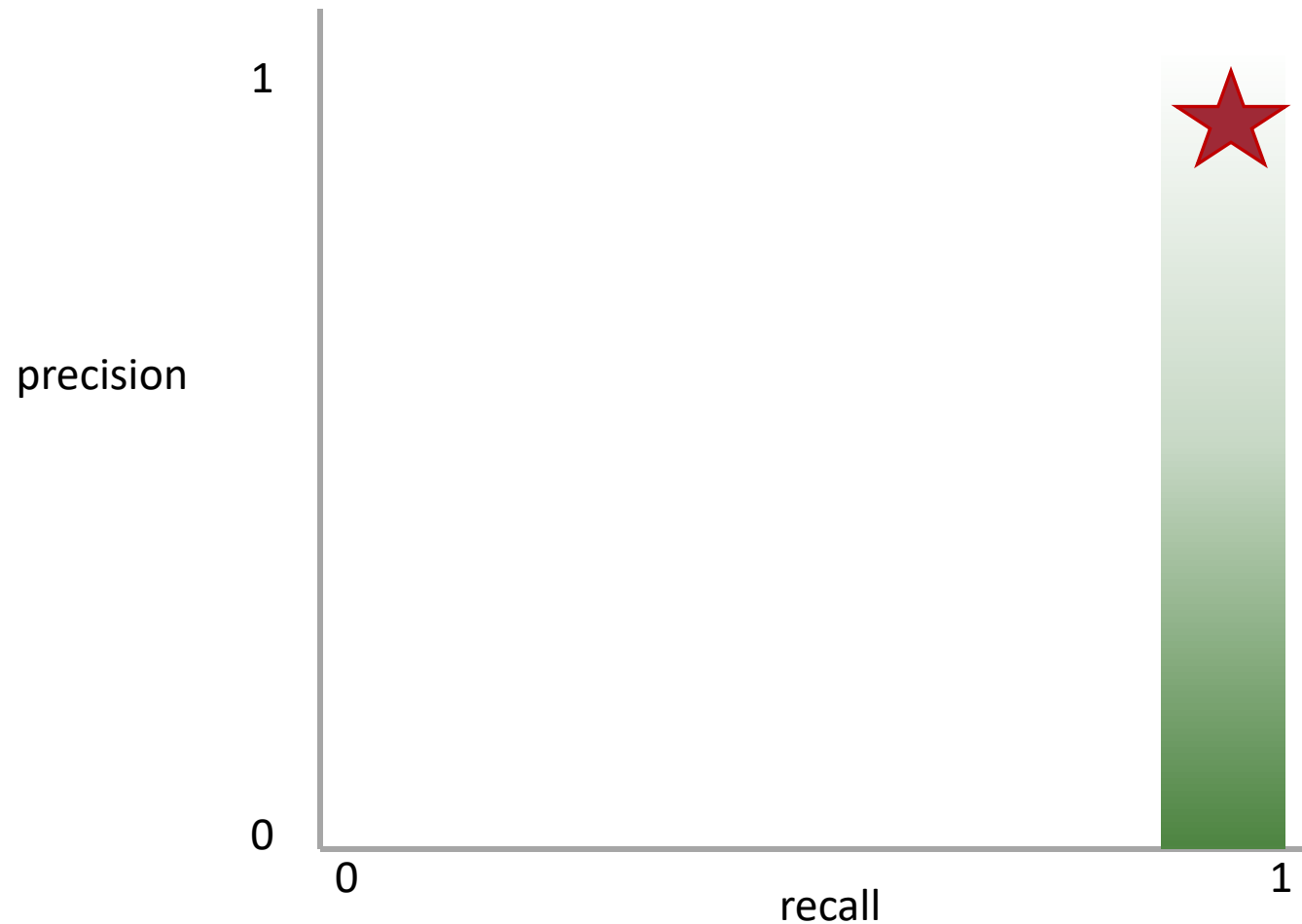


# Precision and Recall Present a Tradeoff

Q: Where do you want your ideal model ?



# Precision and Recall Present a Tradeoff

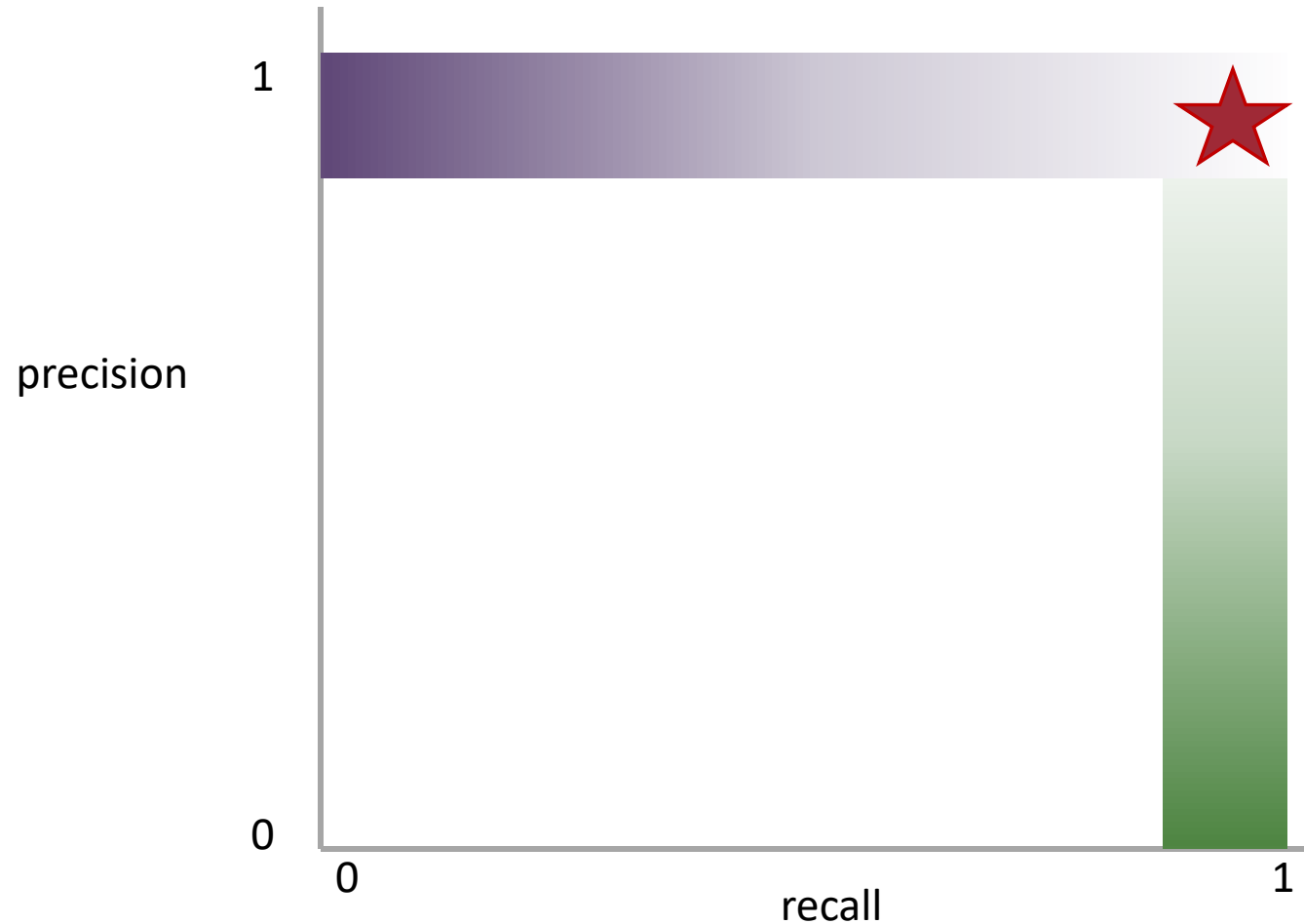


Q: Where do you want your ideal **model** ?

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

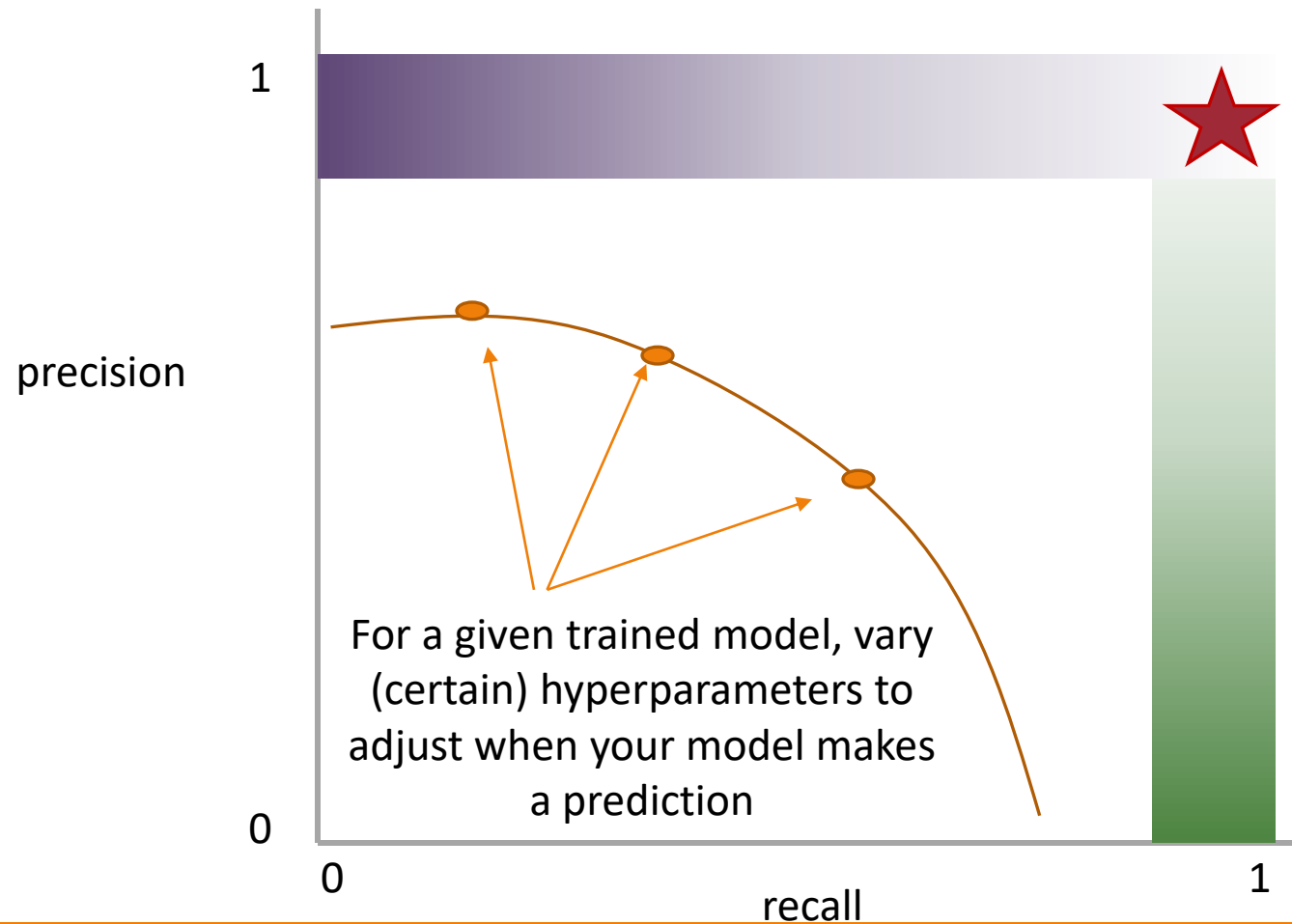


Q: Where do you want your ideal model ?

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



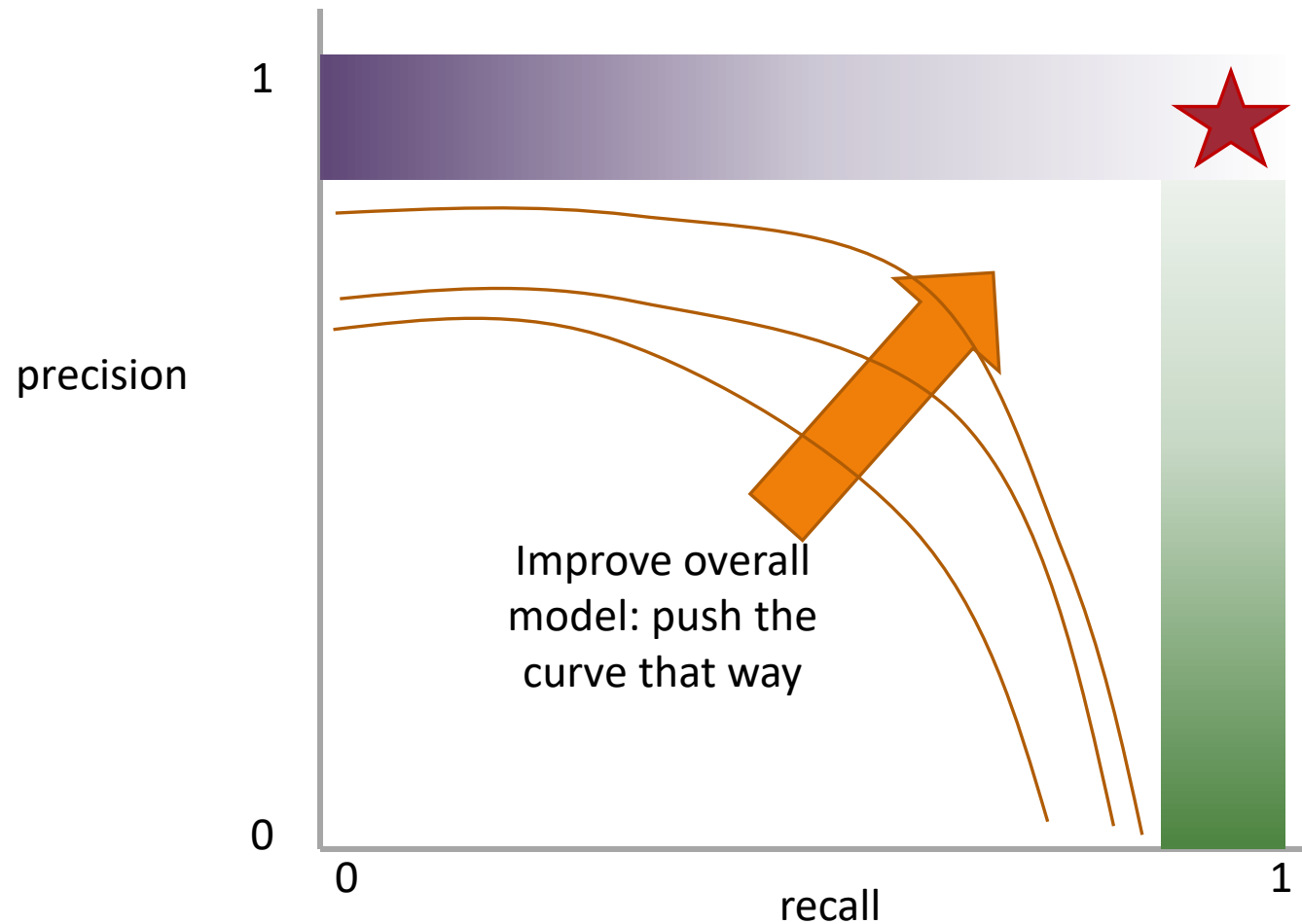
Q: Where do you want your ideal model ?

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 ?

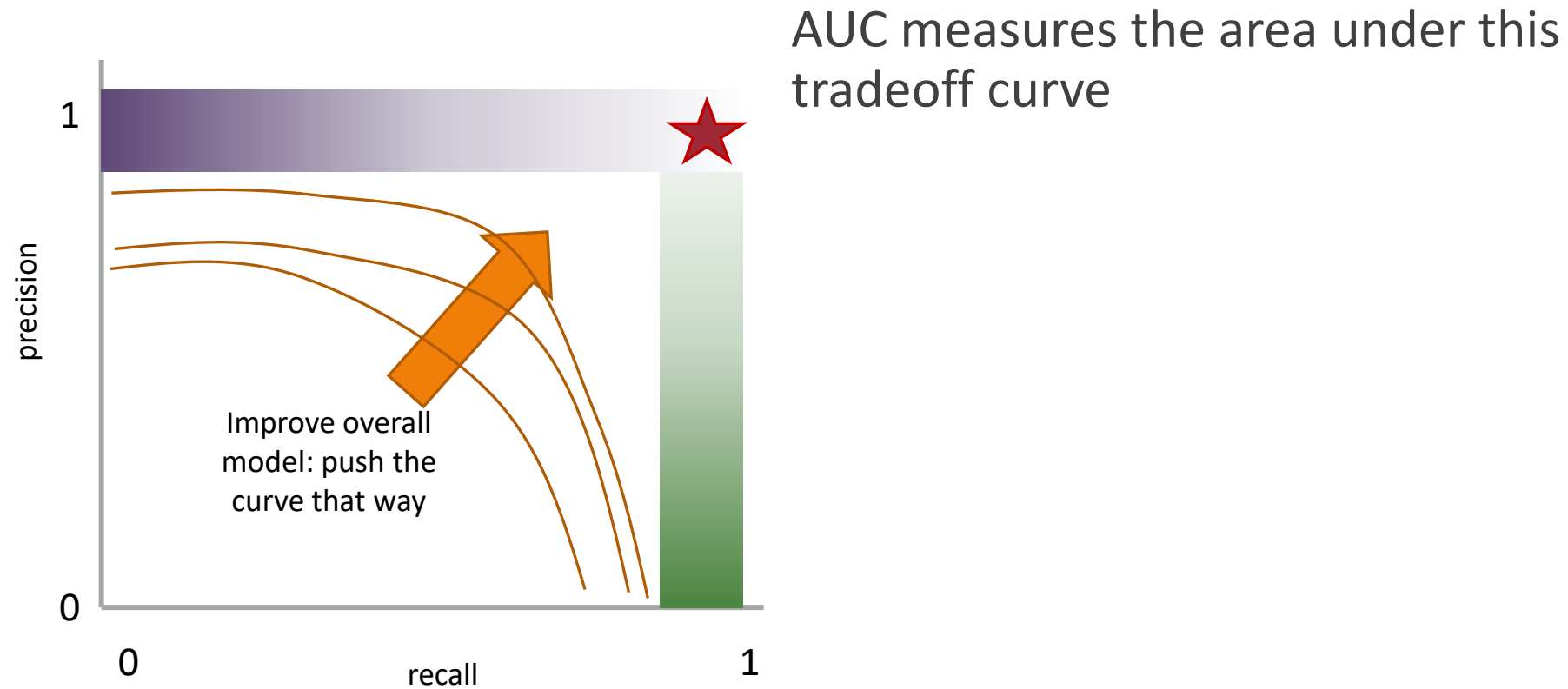
Q: You have a model that always identifies correct instances. Where on this graph is it?

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

Idea: measure the tradeoff between precision and recall



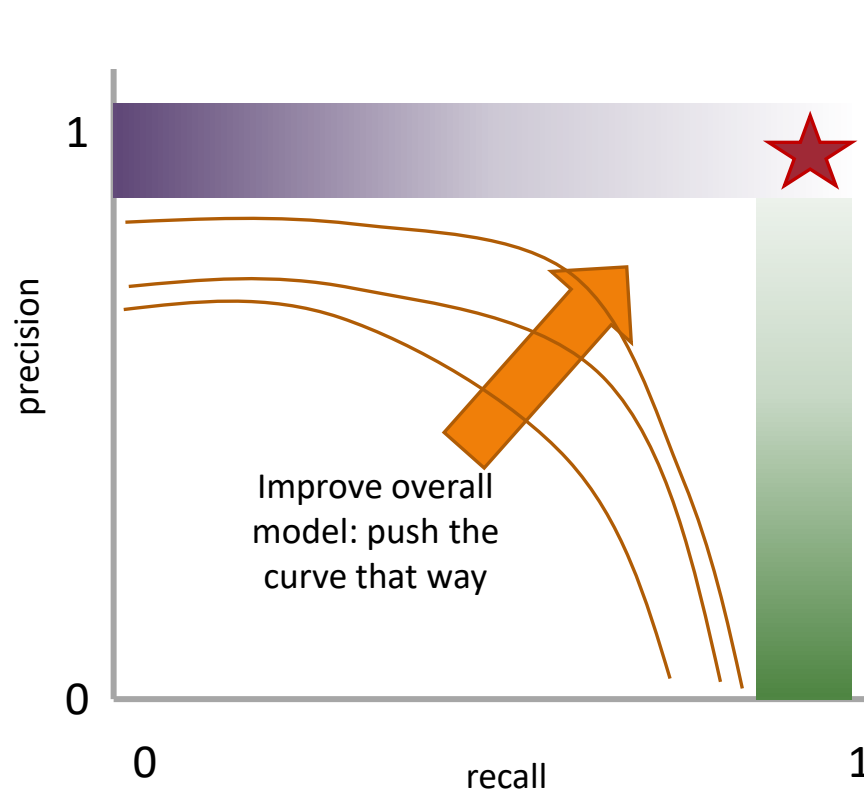
# Measure this Tradeoff: Area Under the Curve (AUC)



Min AUC: 0 🙄

Max AUC: 1 😄

# Measure this Tradeoff: Area Under the Curve (AUC)



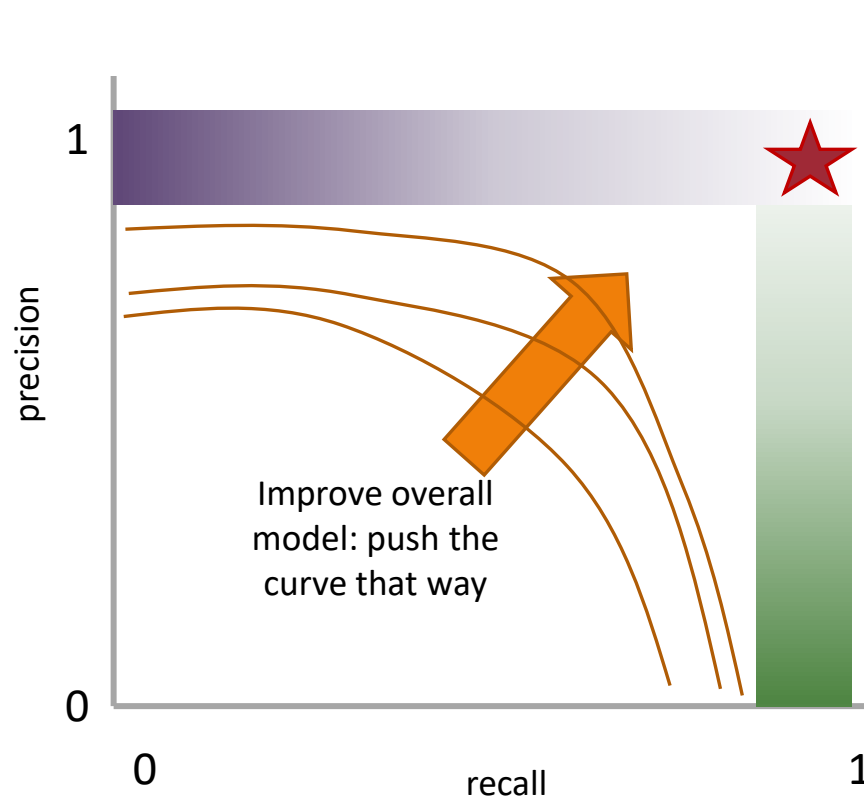
AUC measures the area under this tradeoff curve

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

Min AUC: 0 🙄  
Max AUC: 1 😄

# Measure this Tradeoff: Area Under the Curve (AUC)



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)

Min AUC: 0 🙄  
Max AUC: 1 😄

**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.

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

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

**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.

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

when to prefer  
macroaveraging?

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

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

when to prefer  
microaveraging?



# But how do we compute stats for multiple classes?

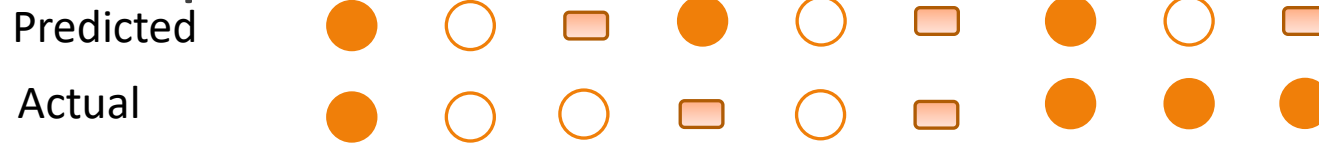
---

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

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

# 1. Compute “one-vs-all” 2x2 tables

Predicted




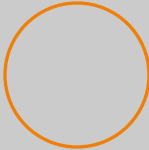


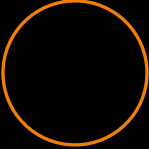

Actual



Look for ●	Actually Target	Actually Not Target	Look for ○	Actually Target	Actually Not Target
Selected/G uessed	2	1	Selected/G uessed	2	1
Not select/not guessed	2	4	Not select/not guessed	1	5

Look for ◻	Actually Target	Actually Not Target
Selected/G uessed	1	2
Not select/not guessed	1	5

## 2. Generalizing the 2-by-2 contingency table

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





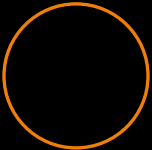

## 2. Generalizing the 2-by-2 contingency table

Predicted



Actual



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


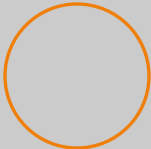


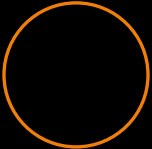

## 2. Generalizing the 2-by-2 contingency table

Predicted



Actual



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





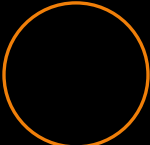

## 2. Generalizing the 2-by-2 contingency table

Predicted



Actual



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

How do you compute  $TP_{\bullet}$ ?  $\frac{TP}{TP + FN}$





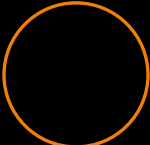

## 2. Generalizing the 2-by-2 contingency table

Predicted



Actual



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

How do you compute  $TP_{\bullet}$ ?




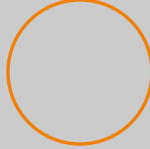


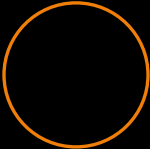

## 2. Generalizing the 2-by-2 contingency table

Predicted



Actual



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

How do you compute  $FN$ ?





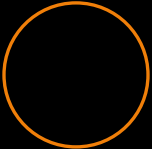

## 2. Generalizing the 2-by-2 contingency table

Predicted



Actual



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

How do you compute  $FN$ ?


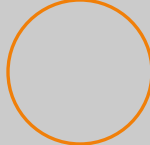


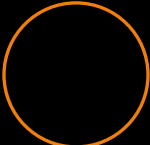

## 2. Generalizing the 2-by-2 contingency table

Predicted



Actual



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

How do you compute  $FP_{\text{white circle}}$ ?





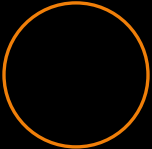

## 2. Generalizing the 2-by-2 contingency table

Predicted




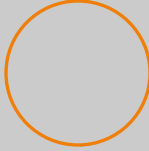


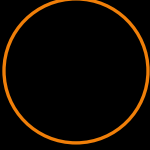

Actual



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


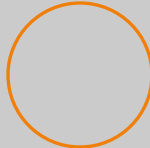


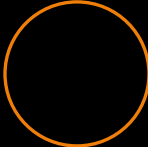

How do you compute  $FP_{\text{white circle}}$ ?

# Generalizing the 2-by-2 contingency table


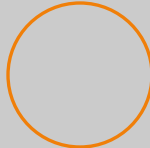


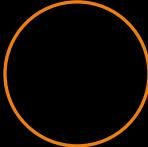

		Correct Value		
				
Guessed Value		#	#	#
		#	#	#
		#	#	#

This is also called a **Confusion Matrix**


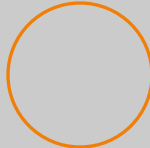


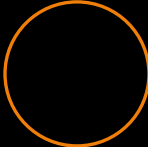

# Generalizing the 2-by-2 contingency table

Q: Is this a good result?		Correct Value		
				
Guessed Value		80	9	11
		7	86	7
		2	8	9

# Generalizing the 2-by-2 contingency table

Q: Is this a good result?		Correct Value		
				
Guessed Value		30	40	30
		25	30	50
		30	35	35

# Generalizing the 2-by-2 contingency table

Q: Is this a good result?		Correct Value		
				
Guessed Value		7	3	90
		4	8	88
		3	7	90