Introduction to Machine Learning Applications

Spring 2023

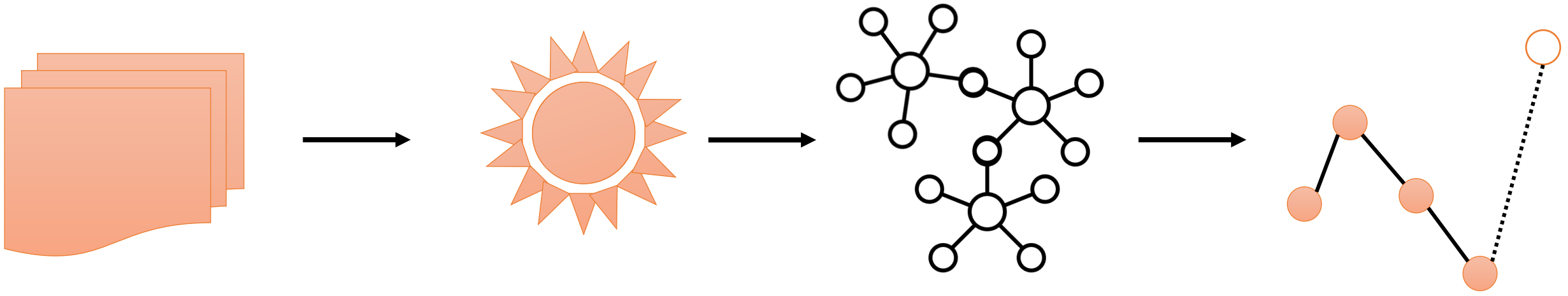
Model evaluation

Minor Gordon

gordom6@rpi.edu

Rensselaer

A standard learning pipeline



Evaluating classification

Overall

- Confusion Matrix
- Accuracy/Precision/Recall

Alternate models

- Area under ROC Curve
- Custom

Evaluating classification

CONFUSION MATRIX

"confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one."
https://en.wikipedia.org/wiki/Confusion_matrix

	Predicted Class		
		True	False
Actual Class	True	True positive (tp)	False Negative (fn)
	False	False Positive (fp)	True Negative (tn)

Confusion Matrix

- A table that is often used to describe the performance of a classification model on a set of test data.
- This allows the visualization of the algorithm's performance.

		Actual Class	
		Class = 1	**Class = 0**
Predicted Class	**Class = 1**	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

Confusion matrix

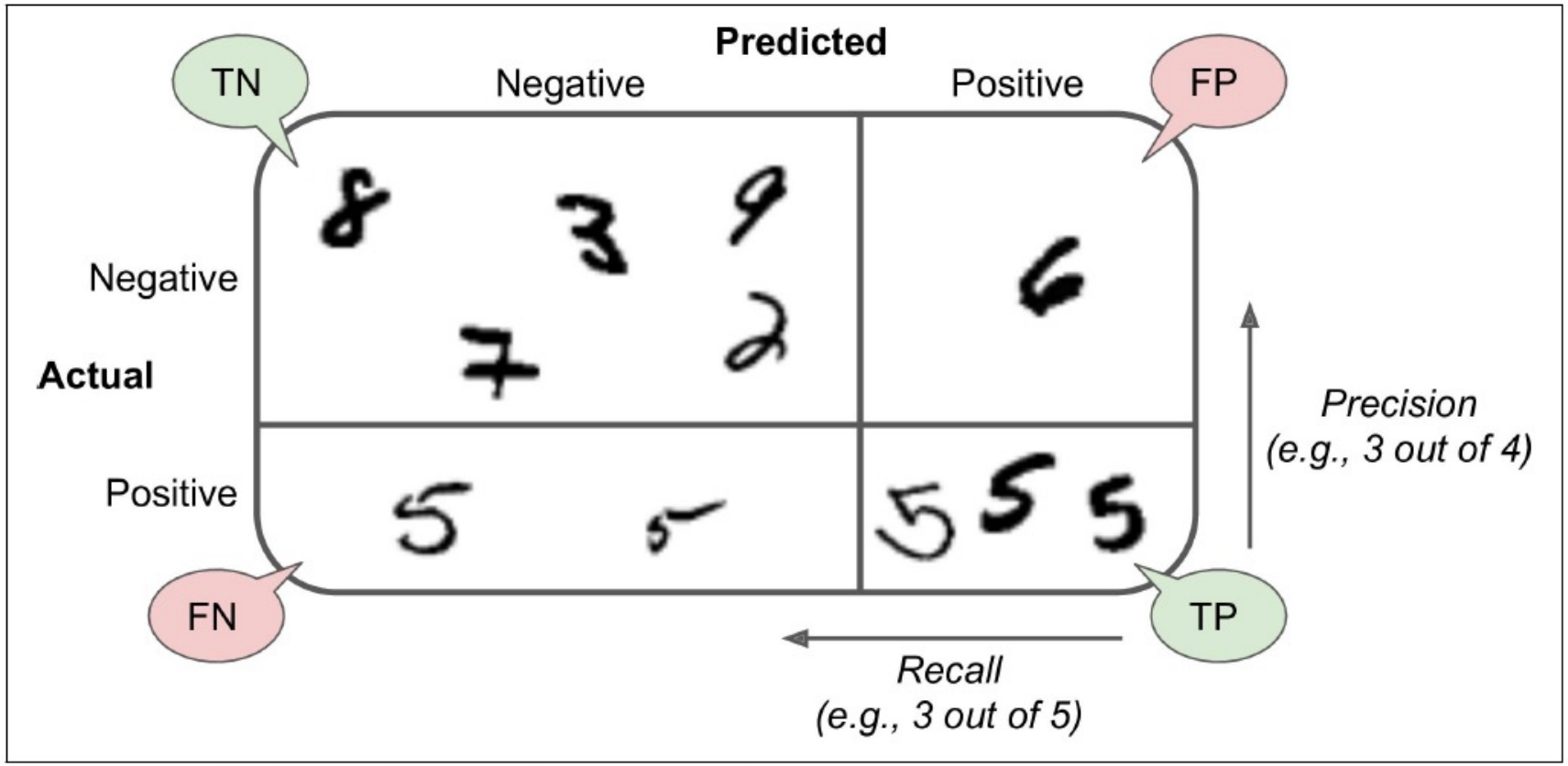


Figure 3-2. An illustrated confusion matrix

Accuracy

Accuracy = (tp + tn) / (P+N)
= (tp + tn)/(tp + fp + tn + fn)

Both classes interesting and not severely unbalanced (why?).

	Predicted Class		
		True	False
Actual Class	True	True positive (tp)	False Negative (fn)
	False	False Positive (fp)	True Negative (tn)

		Actual Class	
		Class = 1	**Class = 0**
Predicted Class	**Class = 1**	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

- f_{11} – True Positive
- f_{10} – False Positive – Type I error
- f_{01} – False Negative – Type II error
- f_{00} – True Negative

		Actual Class	
		Class = 1	**Class = 0**
Predicted Class	**Class = 1**	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

- f_{11} – True Positive
- f_{10} – False Positive – Type I error
- f_{01} – False Negative – Type II error
- f_{00} – True Negative

$$Accuracy = \frac{(f_{11} + f_{00})}{(f_{11} + f_{10} + f_{01} + f_{00})}$$

		Actual Class	
		Class = 1	**Class = 0**
Predicted Class	**Class = 1**	*10*	*5*
	Class = 0	*5*	*10*

Compute the accuracy:

How many are truly labeled
Total data points that you have:
Accuracy

Sensitivity/Recall/Hit Rate/True Positive Rate (TPR)

Sensitivity/Recall = tp/P
= (tp)/(tp + fn)

(Positive class more interesting.)

	Predicted Class		
		True	False
Actual Class	True	True positive (tp)	False Negative (fn)
	False	False Positive (fp)	True Negative (tn)

Recall

How many relevant items are selected?

$$Recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

$$Recall = \frac{f_{11}}{(f_{01} + f_{11})}$$

Specificity, Selectivity or True Negative Rate (TNR)

Specificity = tn/N=
= tn/(tn + fp)
(Actual negative class more interesting)

	Predicted Class		
		True	False
Actual Class	True	True positive (tp)	False Negative (fn)
	False	False Positive (fp)	True Negative (tn)

Precision

Precision = tp/(tp + fp)
(Actual positive class more interesting and higher costs of false positives)

	Predicted Class		
		True	False
Actual Class	True	True positive (tp)	False Negative (fn)
	False	False Positive (fp)	True Negative (tn)

Precision

How many selected items are relevant?

$$Precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

$$Precision = \frac{f_{11}}{(f_{10} + f_{11})}$$

Negative Predictive Value

Negative Predictive Value = tn/(tn + fn)
(Actual neg class more interesting and higher costs of false negatives)

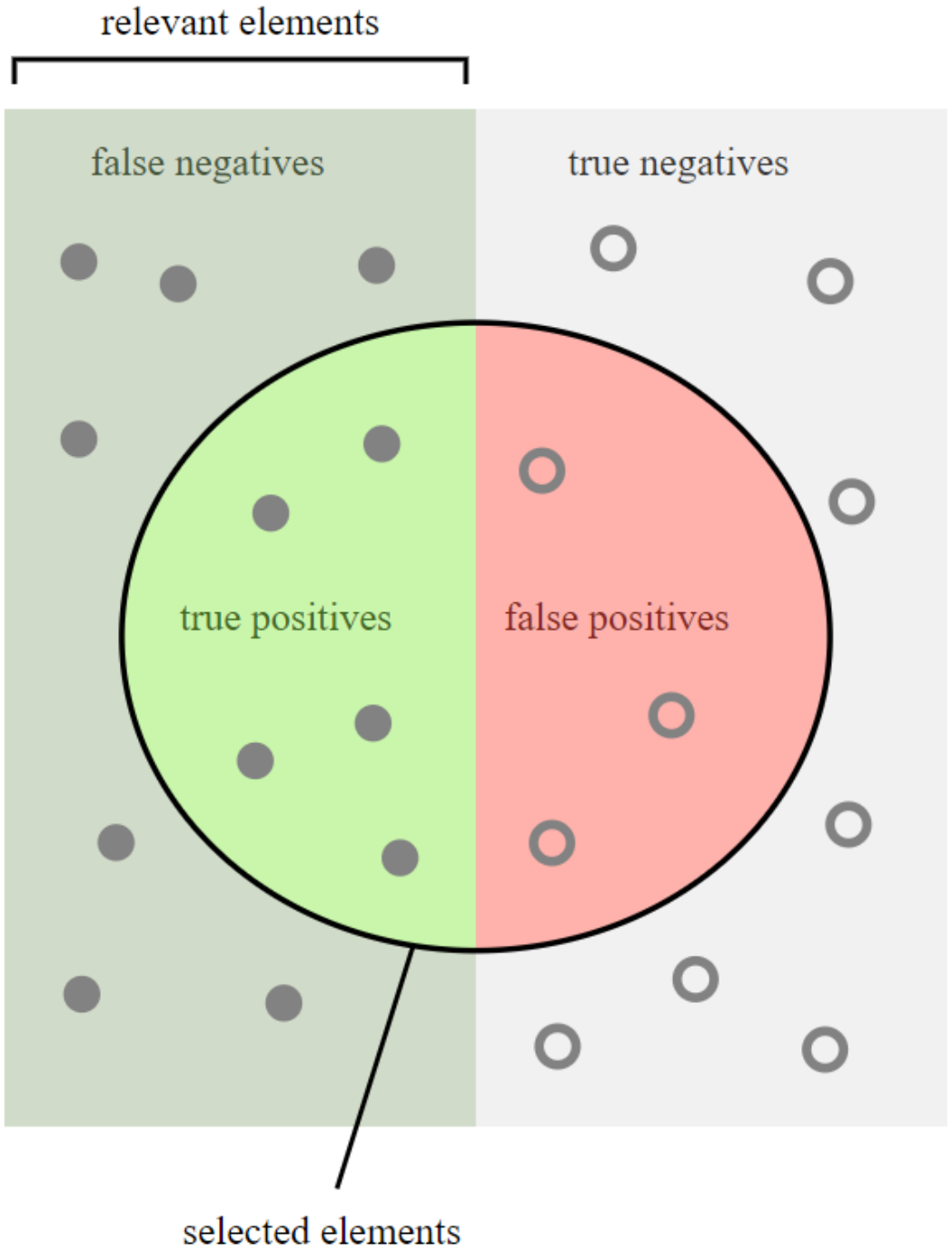
	Predicted Class		
		True	False
Actual Class	True	True positive (tp)	False Negative (fn)
	False	False Positive (fp)	True Negative (tn)

		Actual Class	
		Class = 1	**Class = 0**
Predicted Class	**Class = 1**	f_{11}	f_{10}
	Class = 0	f_{01}	f_{00}

- f_{11} – True Positive
- f_{10} – False Positive – Type I error
- f_{01} – False Negative – Type II error
- f_{00} – True Negative

Precision: How many selected items are relevant?

Recall: How many relevant items are selected?



How many selected items are relevant?

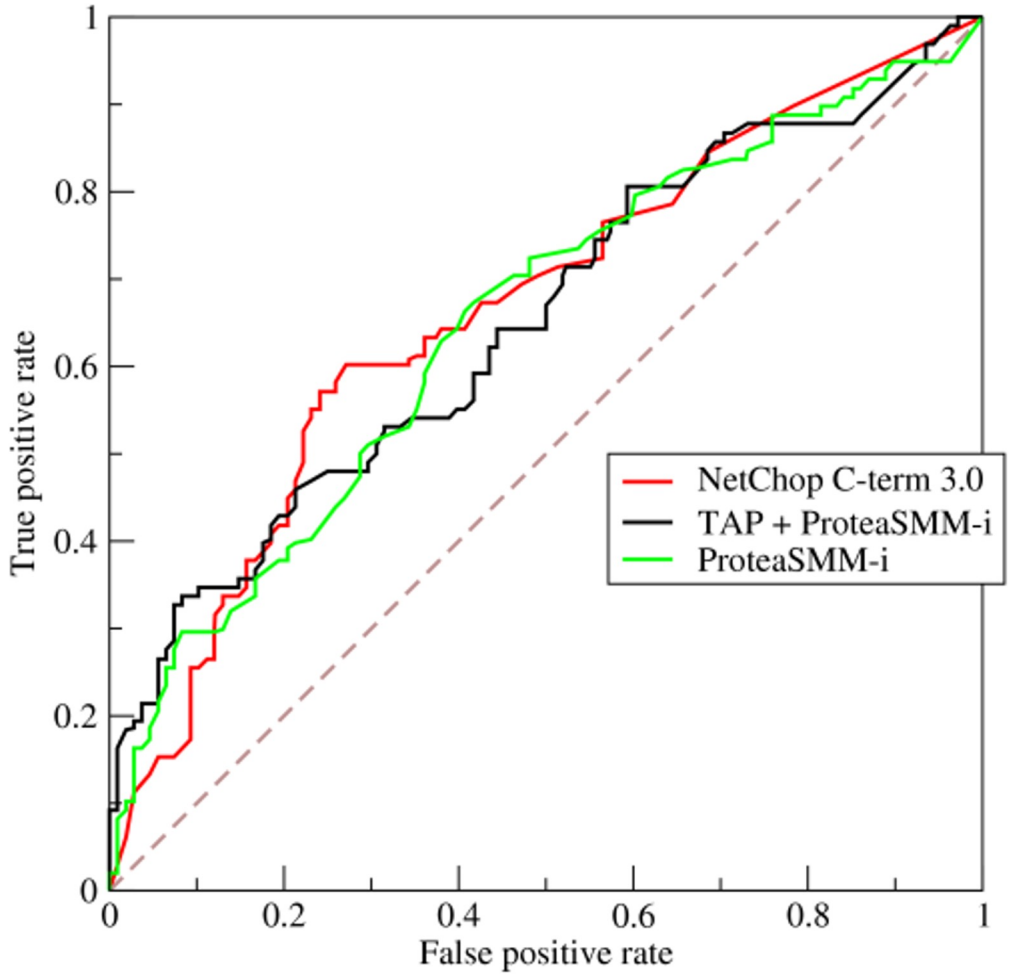
$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Credits: https://en.wikipedia.org/wiki/Precision_and_recall

If we want to balance the true positive or false negative.

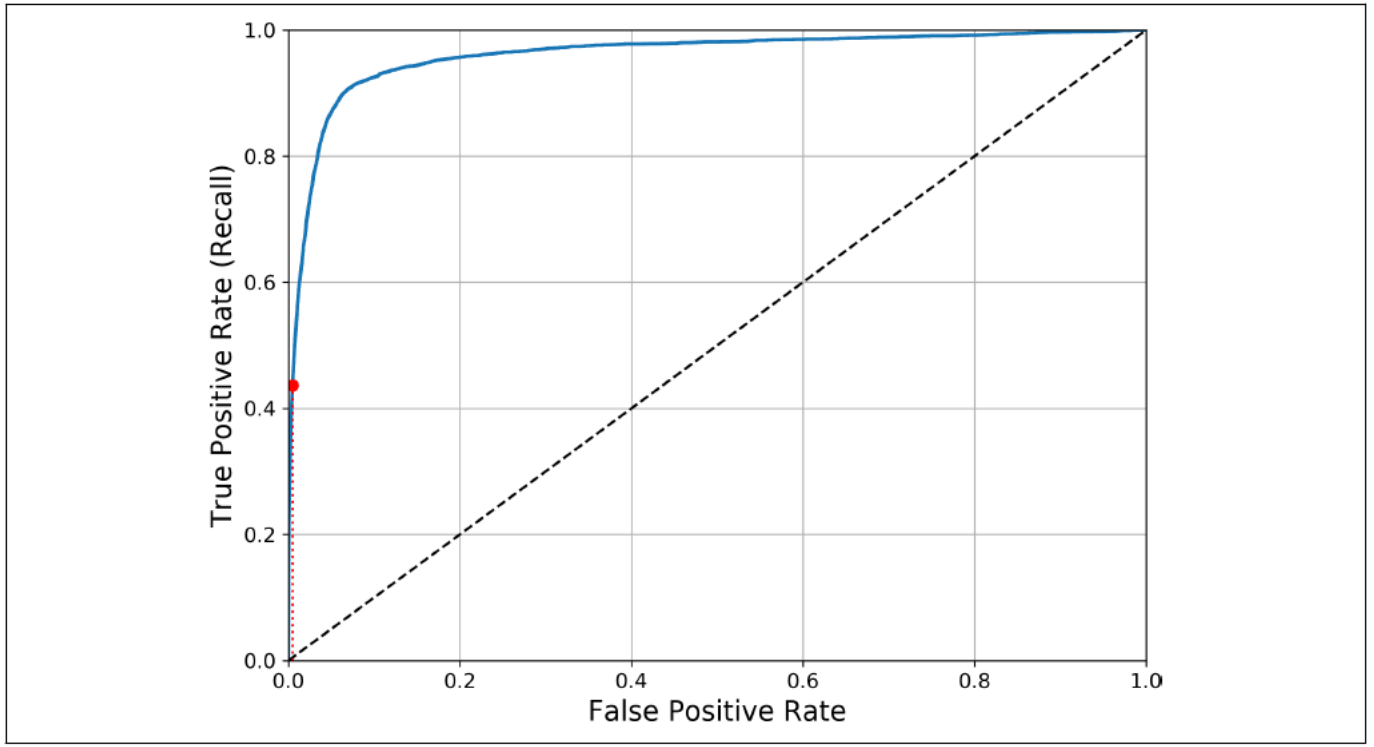


"Receiver operating characteristic (**ROC**), or **ROC curve**, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied."

- Wikipedia

Area under ROC Curve can be used as a performance metric.

ROC


True Positive Rate (Recall)
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

F-measure

Better measure that considers the harmonic mean of *precision* and *recall*

$$f - measure = \frac{2*(precision*recall)}{(precision+recall)}$$

$$f1score = \frac{2 * precision * recall}{(precision + recall)}$$

Compute precision, recall and f-measure

tp = 8

fp = 4

fn = 2

tn = 6

		Actual Class	
		True	**False**
Predicted class	**True**	8	*4*
	False	2	6

Evaluating classification

- When goal is to identify rare outcomes, best model may have lower accuracy
- Must ask: what is the value of a false positive, false negative, true positive, true negative
- Tradeoffs in setting the threshold for classification

Precision/Recall Tradeoff

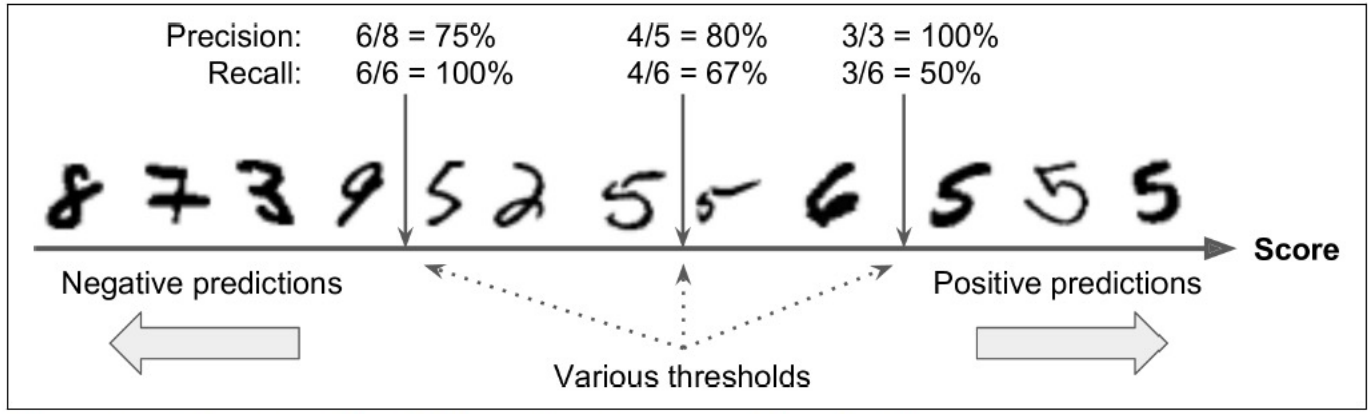
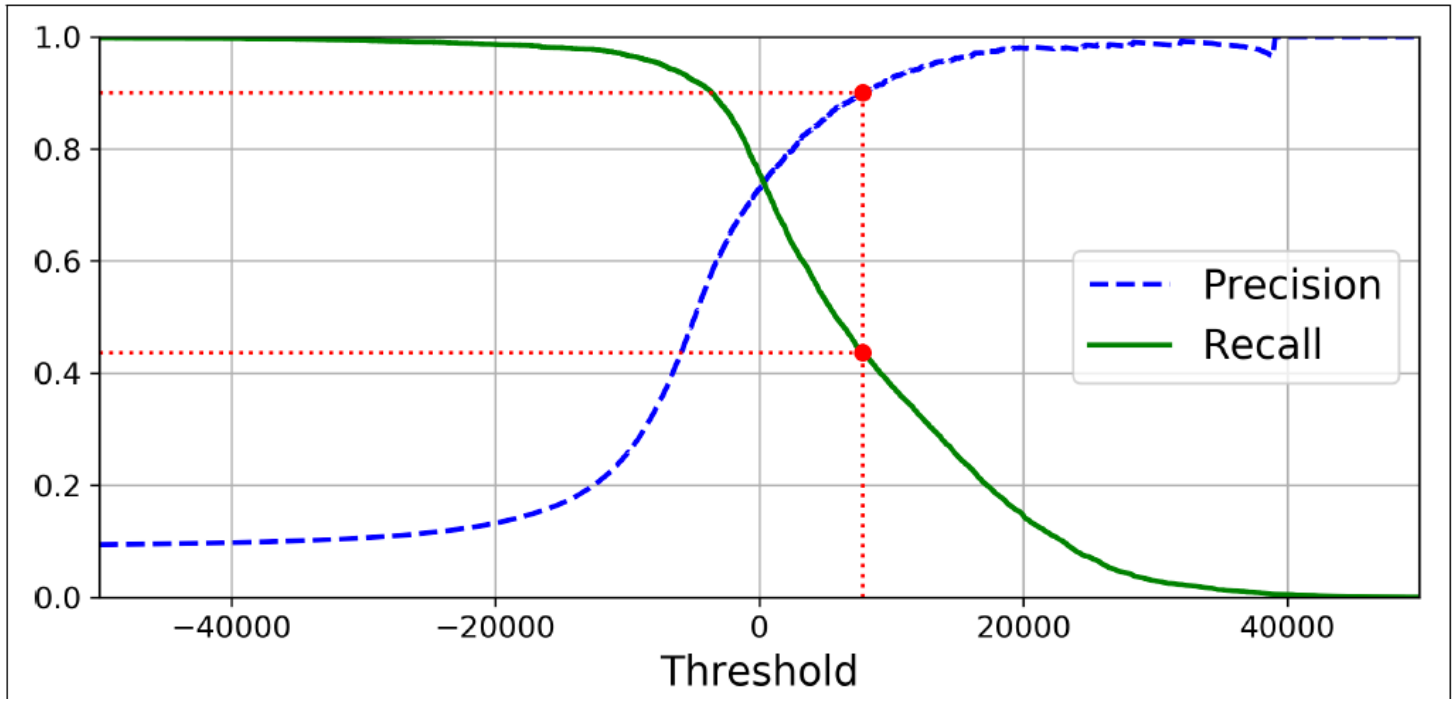


Figure 3-3. Decision threshold and precision/recall tradeoff

- Increasing threshold increases precision but lowers recall
- Decreasing threshold increases recall and lowers precision

Threshold: Precision Recall Curve



- Increasing threshold increases precision but lowers recall
- Decreasing threshold increases recall and lowers precision

Multiclass Outcomes

- Have the choice of training one classifier for each class.
- One-vs-the-rest (OvR) multiclass/multilabel strategy. For each classifier, the class is fitted against all the other classes.
- This is the most commonly used strategy for multiclass classification and is a fair default choice.

10x10 confusion Matrix in MNIST

```
>>> y_train_pred = cross_val_predict(sgd_clf, X_train_scaled, y_train, cv=3)
>>> conf_mx = confusion_matrix(y_train, y_train_pred)
>>> conf_mx
array([[5578,    0,   22,    7,    8,   45,   35,    5,  222,    1],
       [   0, 6410,   35,   26,    4,   44,    4,    8,  198,   13],
       [  28,   27, 5232,  100,   74,   27,   68,   37,  354,   11],
       [  23,   18,  115, 5254,    2,  209,   26,   38,  373,   73],
       [  11,   14,   45,   12, 5219,   11,   33,   26,  299,  172],
       [  26,   16,   31,  173,   54, 4484,   76,   14,  482,   65],
       [  31,   17,   45,    2,   42,   98, 5556,    3,  123,    1],
       [  20,   10,   53,   27,   50,   13,    3, 5696,  173,  220],
       [  17,   64,   47,   91,    3,  125,   24,   11, 5421,   48],
       [  24,   18,   29,   67,  116,   39,    1,  174,  329, 5152]])
```