Metrics for Classification

Learning objectives

Understand the difference between model class predictions versus probability predictions

Learn about the most common error metrics for classification:

- Accuracy and accuracy-based metrics:
  - Confusion matrix
  - Precision and recall
  - Log-loss as a measure that takes the magnitude of uncertainty into account
- Others:
  - ROC curve and maximizing the area under the curve (AUC)

Understand when to apply each metric, particularly the difference between two class and multiclass problems

Classification models predict class and probability

Classification models have two possible types of outputs:

- Class prediction (e.g. whether a patient has a disease or not)
- Probability of being a given class

Not all classification models have a meaningful definition of probability

- Some classification models have a pseudo-probability (e.g. tree models and SVMs)
- In general, getting pseudo-probabilities out of SVMs takes a long time so we don't do it

We judge our models based on the class and probability predictions they make

Sometimes we care only about the class prediction, sometimes we use probabilities as inputs into other downstream models

The most naive metric: Accuracy

Accuracy is % of observations classified correctly

We calculate accuracy for any classification model as:

(% observations correctly classified / all observations)

Accuracy is useful as a first heuristic, but it has shortcomings

Accuracy is % of observations classified correctly

We calculate accuracy for any classification model as:

(% observations correctly classified / all observations)

Accuracy is useful as a first heuristic, but it has shortcomings

Student exercise

(a) Is 95 percent accuracy a good score or not?
(b) Can you name some shortcomings of accuracy as a metric? Think about cases where we're trying to predict highly imbalanced classes.

Accuracy is % of observations classified correctly

We calculate accuracy for any classification model as:

(% observations correctly classified / all observations)

Accuracy is useful as a first heuristic, but it has shortcomings

Say we're trying to predict whether a patient has a disease

In our sample, 99.5% of patients do not have the disease

What's a naive model that would give us high accuracy?

Accuracy is % of observations classified correctly

We calculate accuracy for any classification model as:

(% observations correctly classified / all observations)

Accuracy is useful as a first heuristic, but it has shortcomings

Say we're trying to predict whether a patient has a disease

In our sample, 99.5% of patients do not have the disease

What's a naive model that would give us high accuracy?

If we naively predict "no disease" for every observation, that's 99.5% accuracy!

Accuracy is % of observations classified correctly

We calculate accuracy for any classification model as:

(% observations correctly classified / all observations)

Accuracy is useful as a first heuristic, but it has shortcomings

Say we're trying to predict whether a patient has a disease

In our sample, 99.5% of patients do not have the disease

What's a naive model that would give us high accuracy?

If we naively predict "no disease" for every observation, that's 99.5% accuracy!

This suggests we may want to use other metrics, too

Demystifying the confusion matrix

A confusion matrix is accuracy by class

Predicted

Actual

	# days it rained	# days it was sunny
# days it rained	25	10
# days it was sunny	5	75

A confusion matrix is accuracy by class

Predicted

Actual

	# days it rained	# days it was sunny
# days it rained	25	10
# days it was sunny	5	75

A confusion matrix is used to detail the performance of a classification model.

A confusion matrix is accuracy by class

Predicted

Actual

	# days it rained	# days it was sunny
# days it rained	71%	29%
# days it was sunny	6%	94%

We can normalize our data by turning it into percentages by row.

A confusion matrix is accuracy by class

Predicted

Actual

	Positive class	Negative class
Positive class	True positives	False negatives
Negative class	False positives	True negatives

For those quadrants where our model was correct, we call them true positive/negative. Where our model was wrong, we call them false positive/negative.

A confusion matrix is accuracy by class

Predicted

Actual

	Positive class	Negative class
Positive class	True positives	False negatives
Negative class	False positives	True negatives

Check for understanding: Do the columns/rows sum to 100 percent?

A confusion matrix is accuracy by class

Predicted

Actual

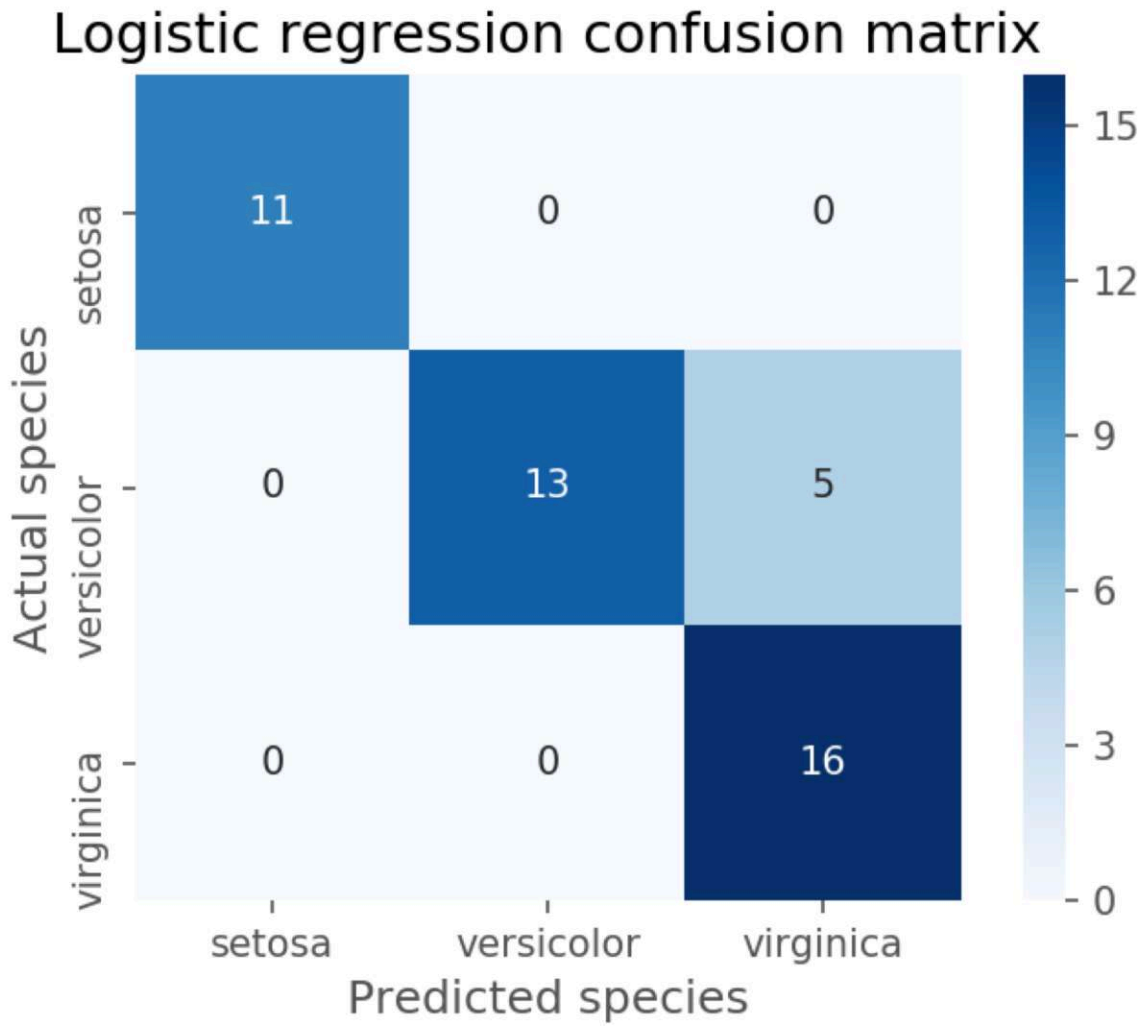
	Positive class	Negative class
Positive class	True positives	False negatives
Negative class	False positives	True negatives

→ sums to 100% when normalized

→ sums to 100% when normalized

In a normalized confusion matrix, the rows are made to sum up to 100%.
This is not a requirement of the confusion matrix, but is a common presentation.

A confusion matrix is useful in multiclass problems



We will tackle this example in the notebook to follow.

Using class probability predictions

Choosing a probability threshold

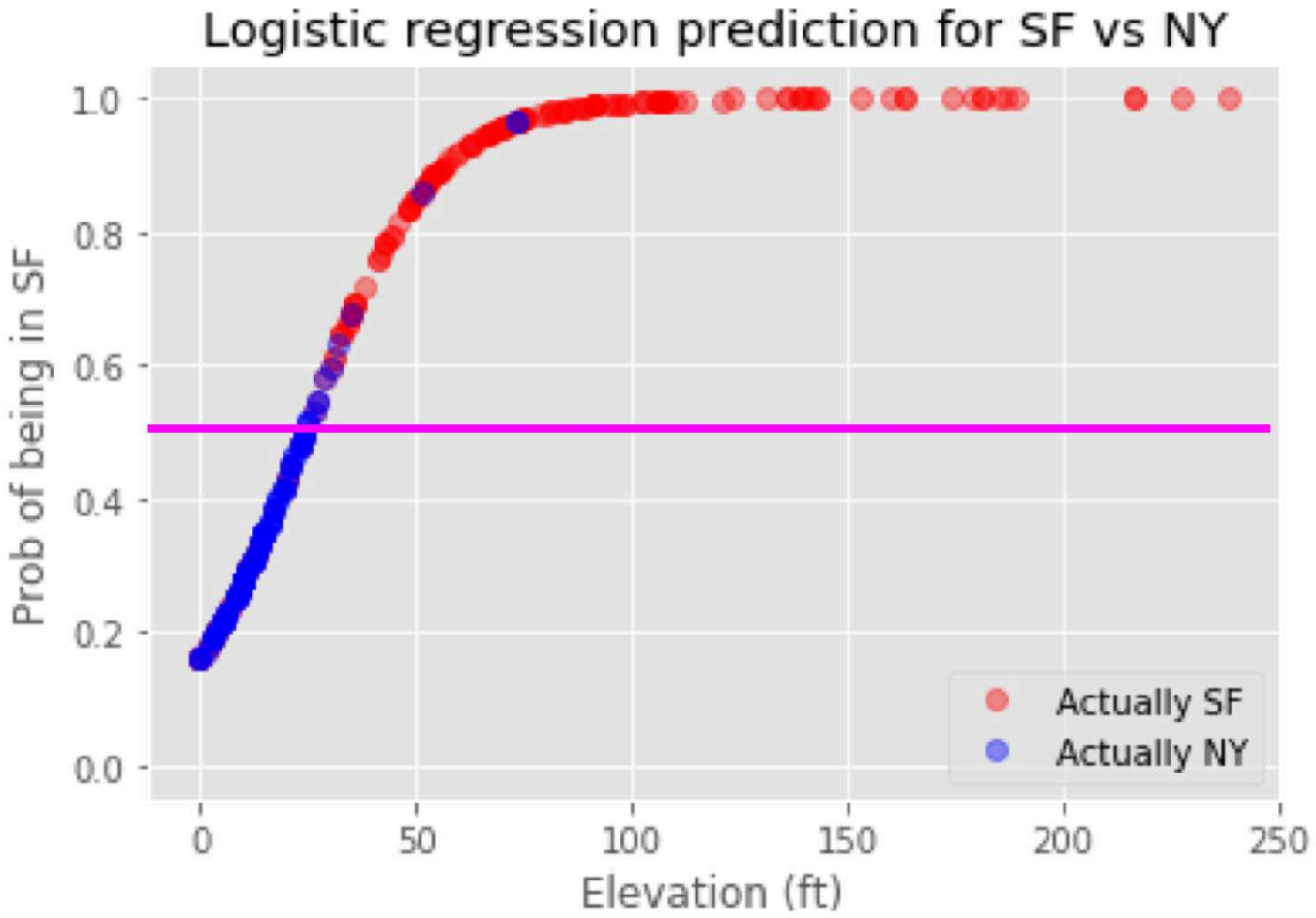
Some models give us probability predictions and not just class predictions

We can use these predicted probabilities to examine how our model's behavior changes as we move the probability threshold

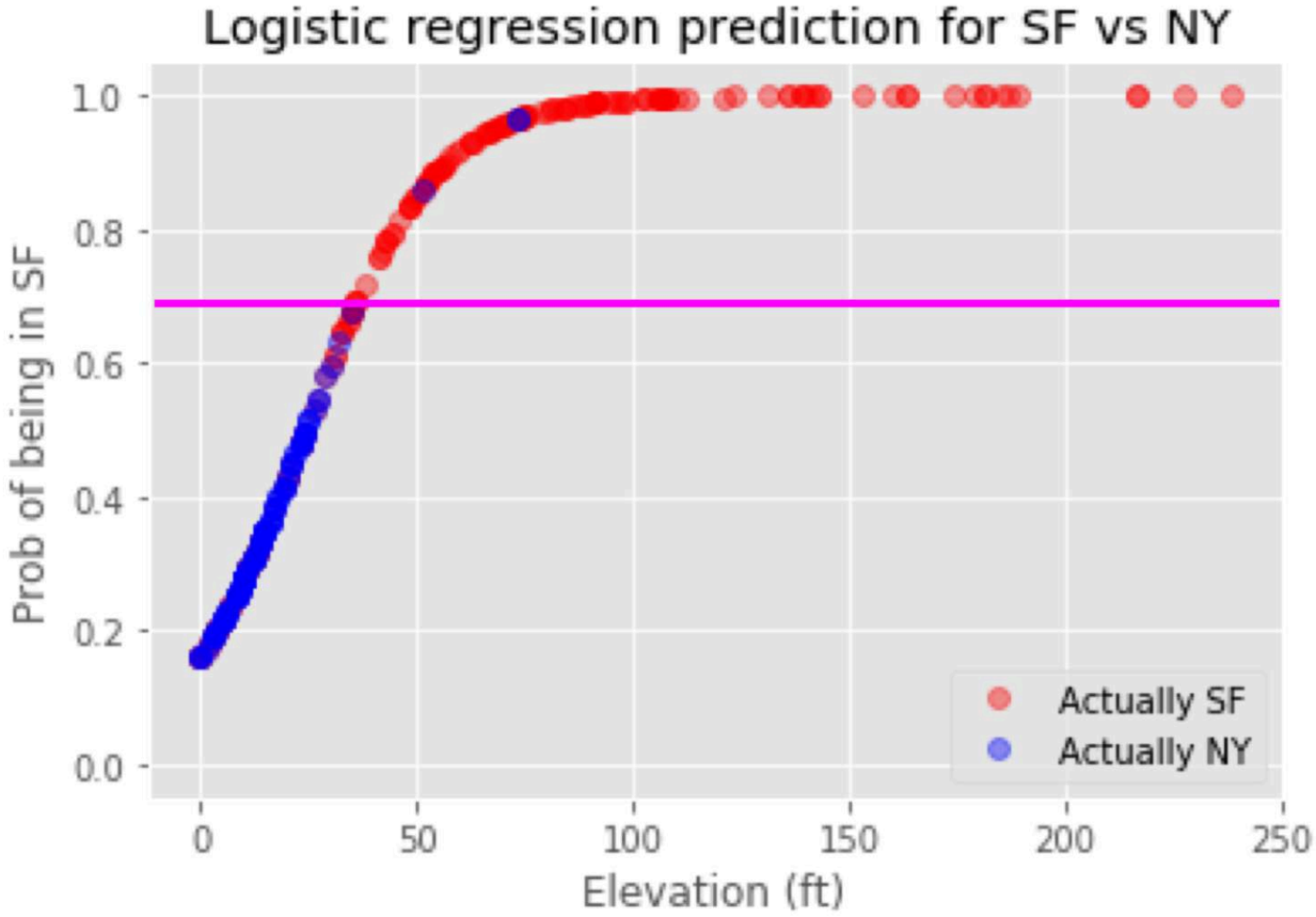
So far we've looked at accuracy, precision and recall derived from a 50% probability threshold (sklearn default)

But, we don't have to take the 50% cutoff, we can choose our own!

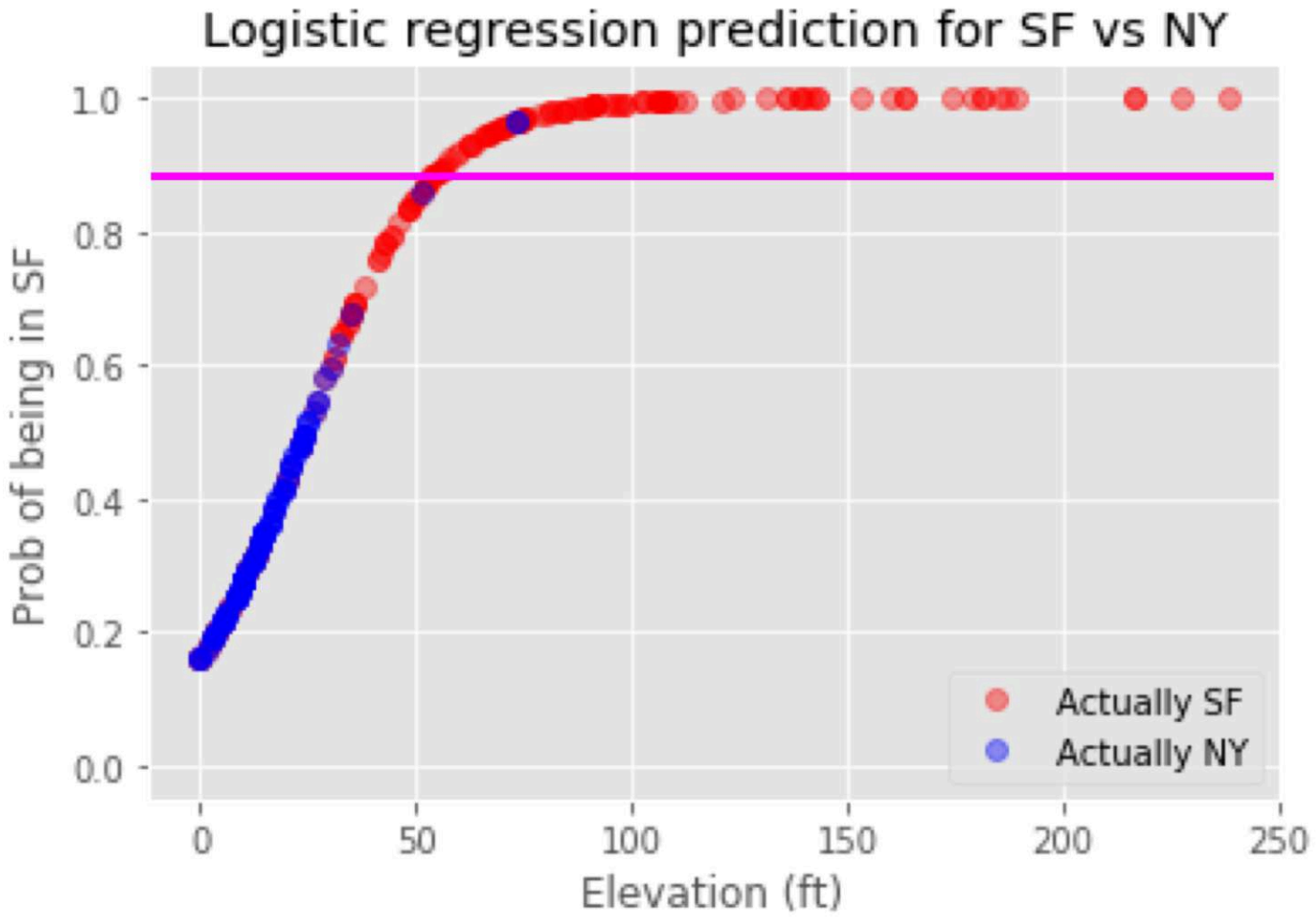
Using a ROC curve to determine probability thresholds



Using a ROC curve to determine probability thresholds



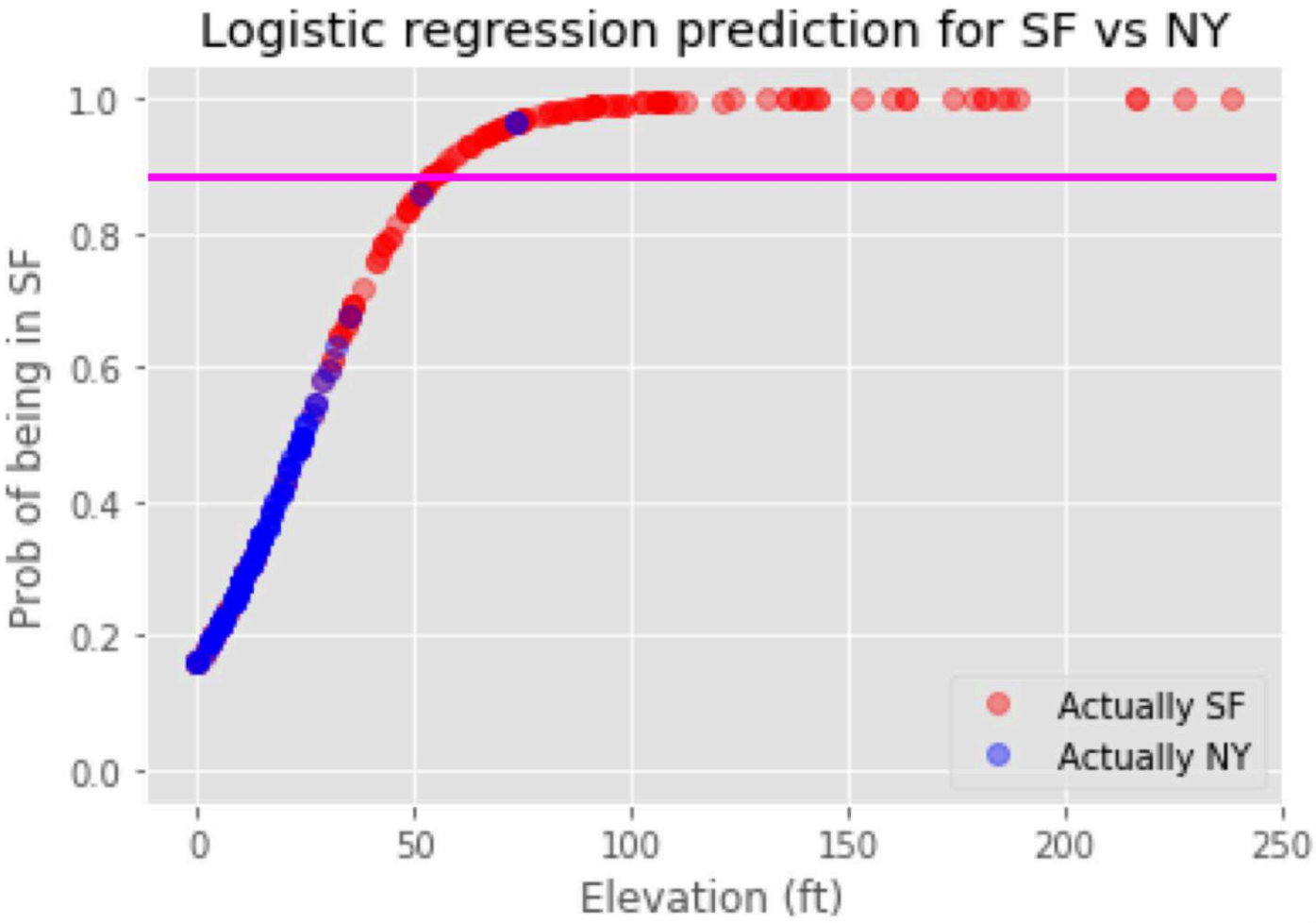
Using a ROC curve to determine probability thresholds



Using a ROC curve to determine probability thresholds

Student exercise:

- As we move the threshold, are we becoming more sure about SF or NYC?
- As we move the threshold, do we have:
  - Lower/higher recall?
  - Lower/higher precision?
  - Lower/higher true positive rate?
    Lower/higher false positive rate?

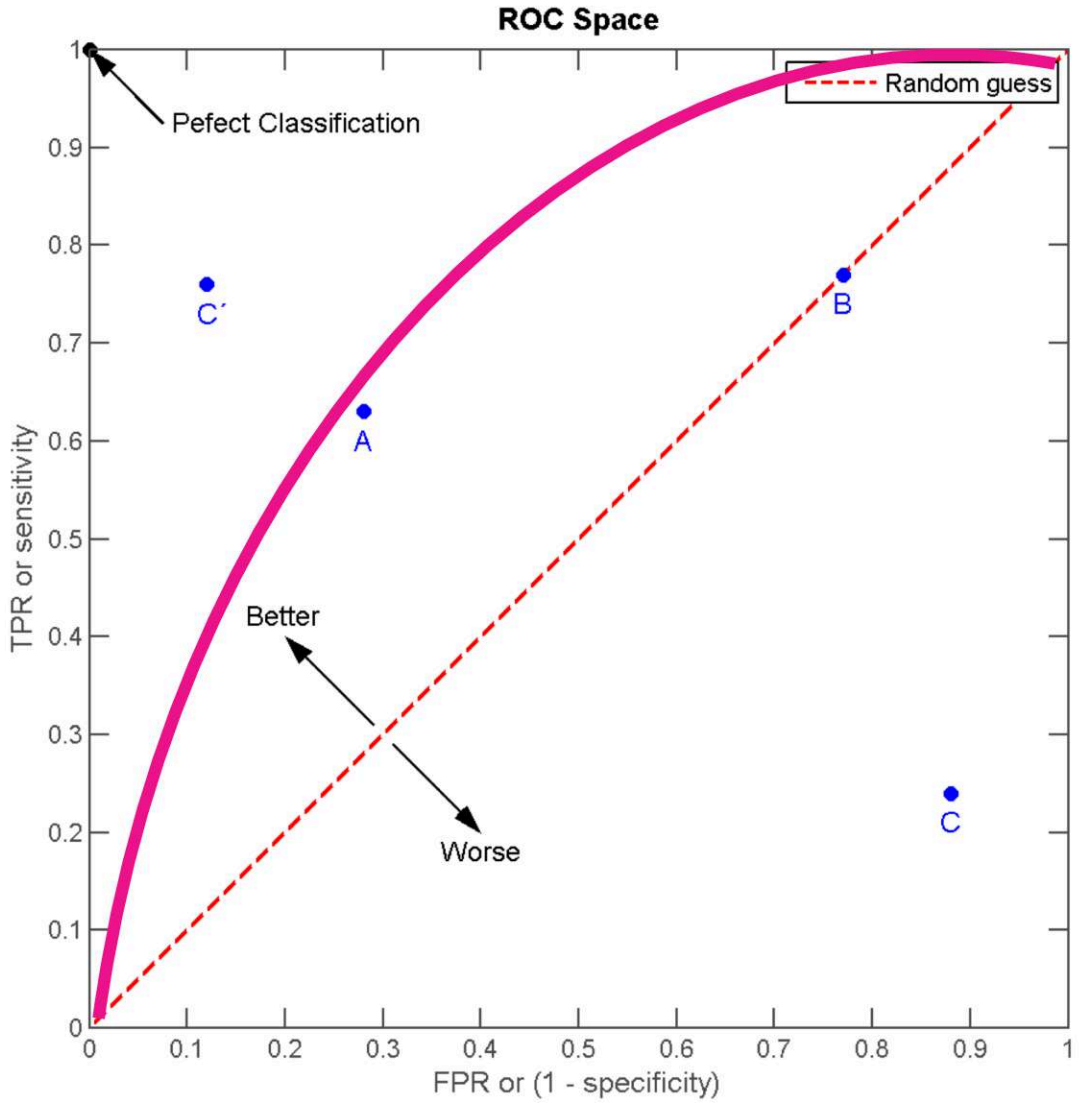


Using a ROC curve to determine probability thresholds

Drawing a ROC curve: change the probability threshold and plot how true positive rate and false positive rate change

Each threshold gives us a new model!

We can plot the ROC curve only for binary cases

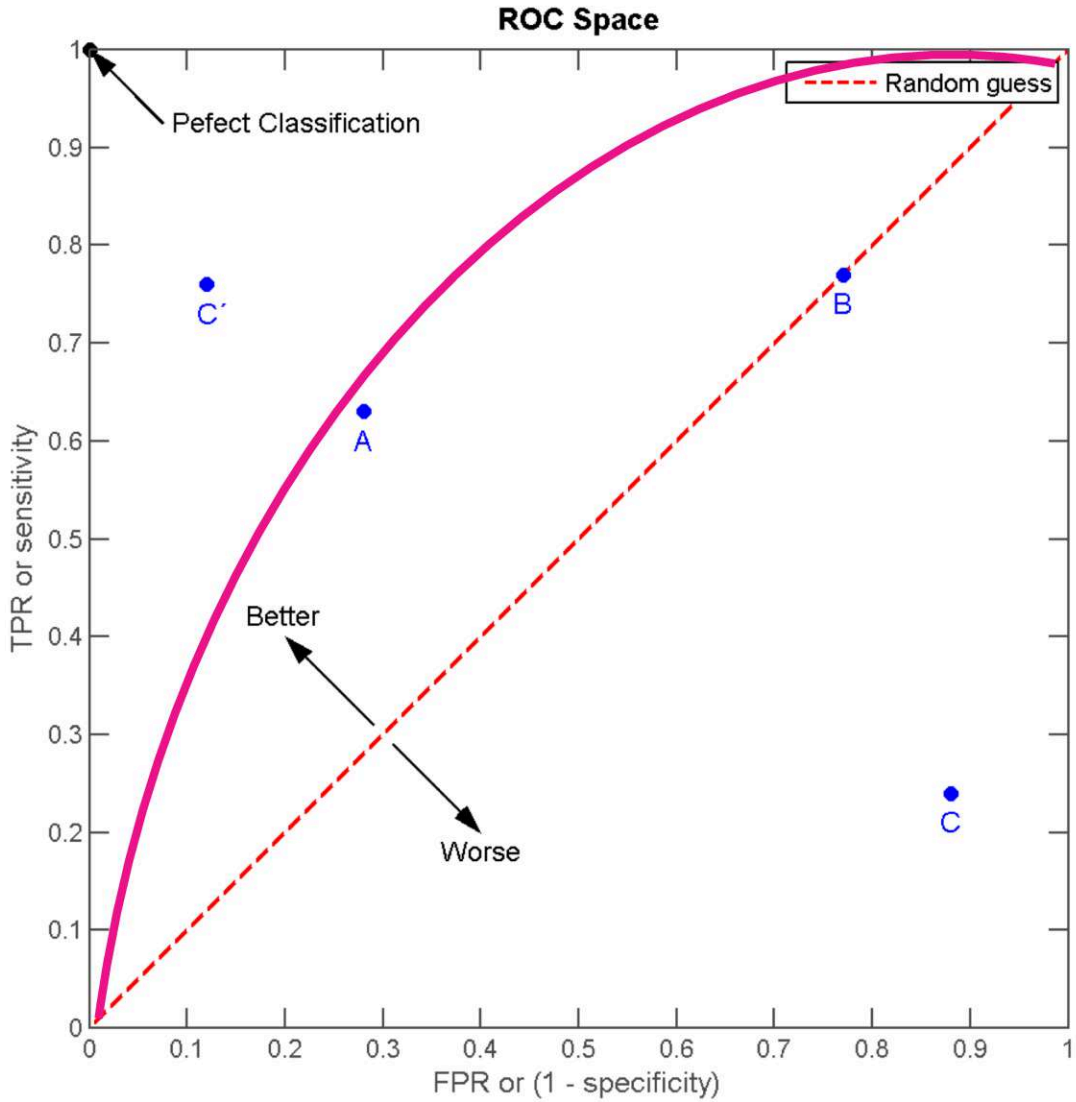


Images from Wikipedia.

Using a ROC curve to determine probability thresholds

Check for understanding:

Which corner represents a higher threshold? Lower threshold?

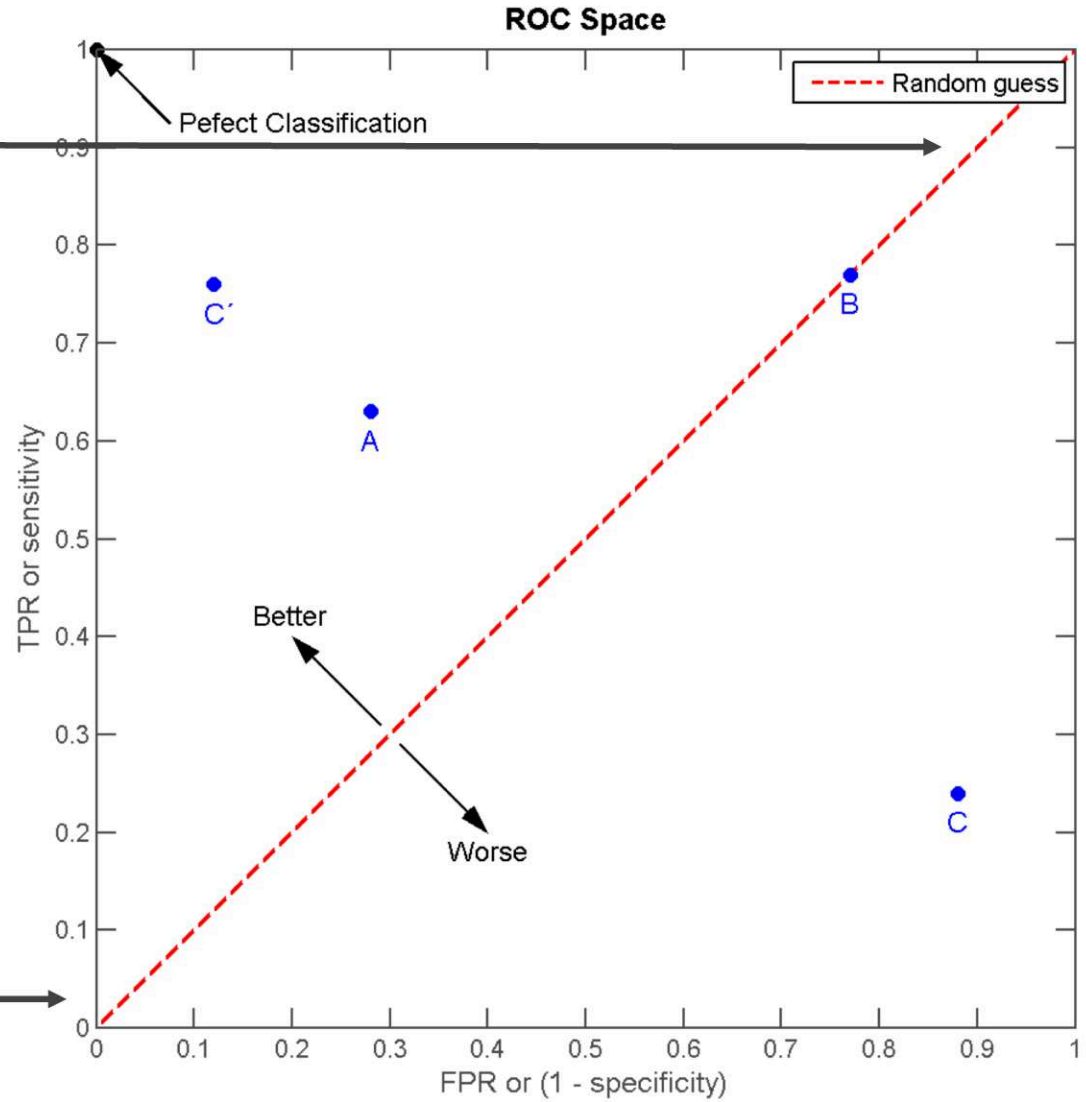


Images from Wikipedia.

Using a ROC curve to determine probability thresholds

- Lower threshold: Better at catching positives. Higher recall, lower precision. Higher true positive rate, higher false positive rate

- Higher threshold: Better at catching negatives. Lower recall, higher precision. Lower true positive rate, lower false positive rate.

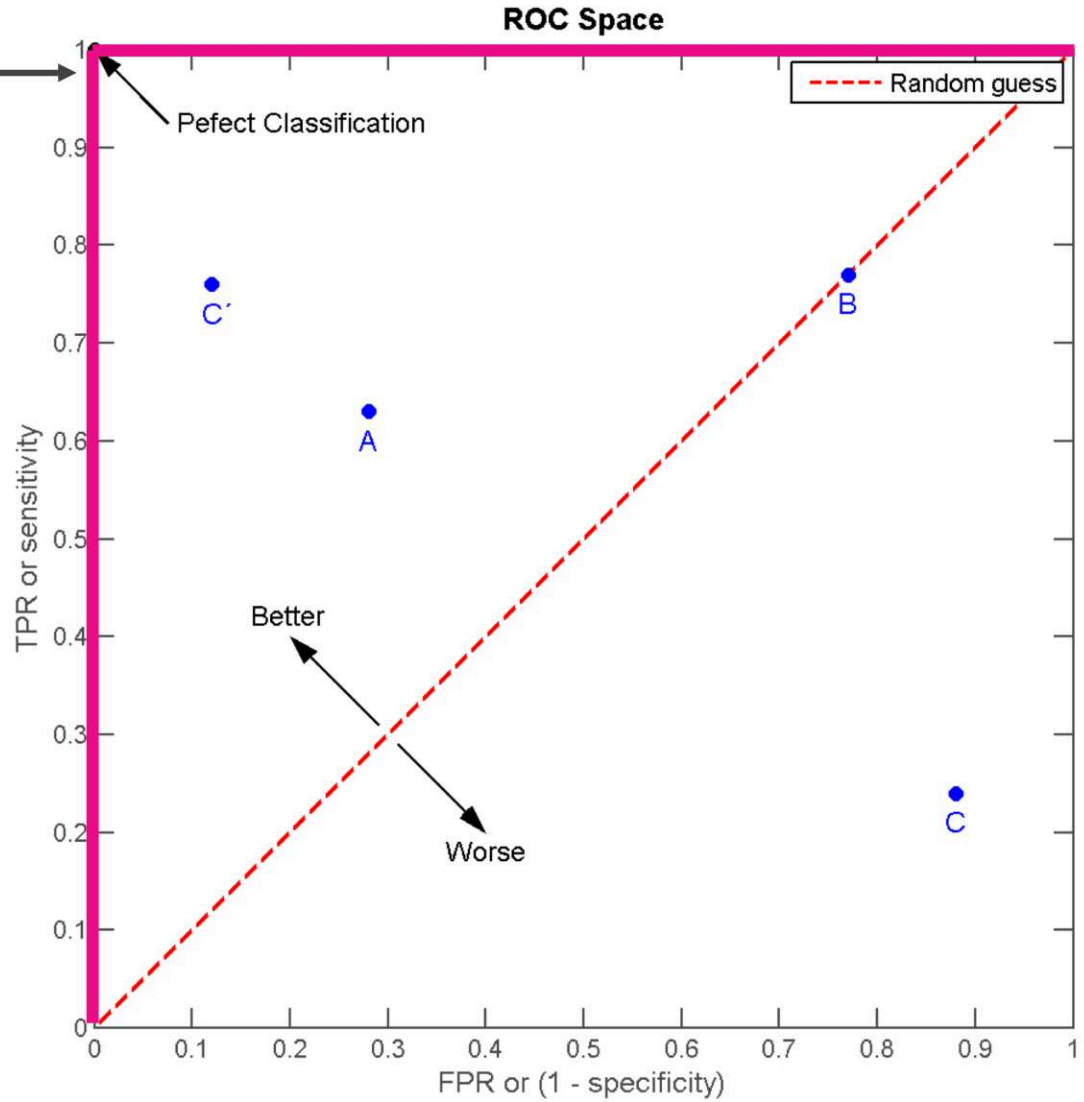


Images from Wikipedia.

The perfect classifier

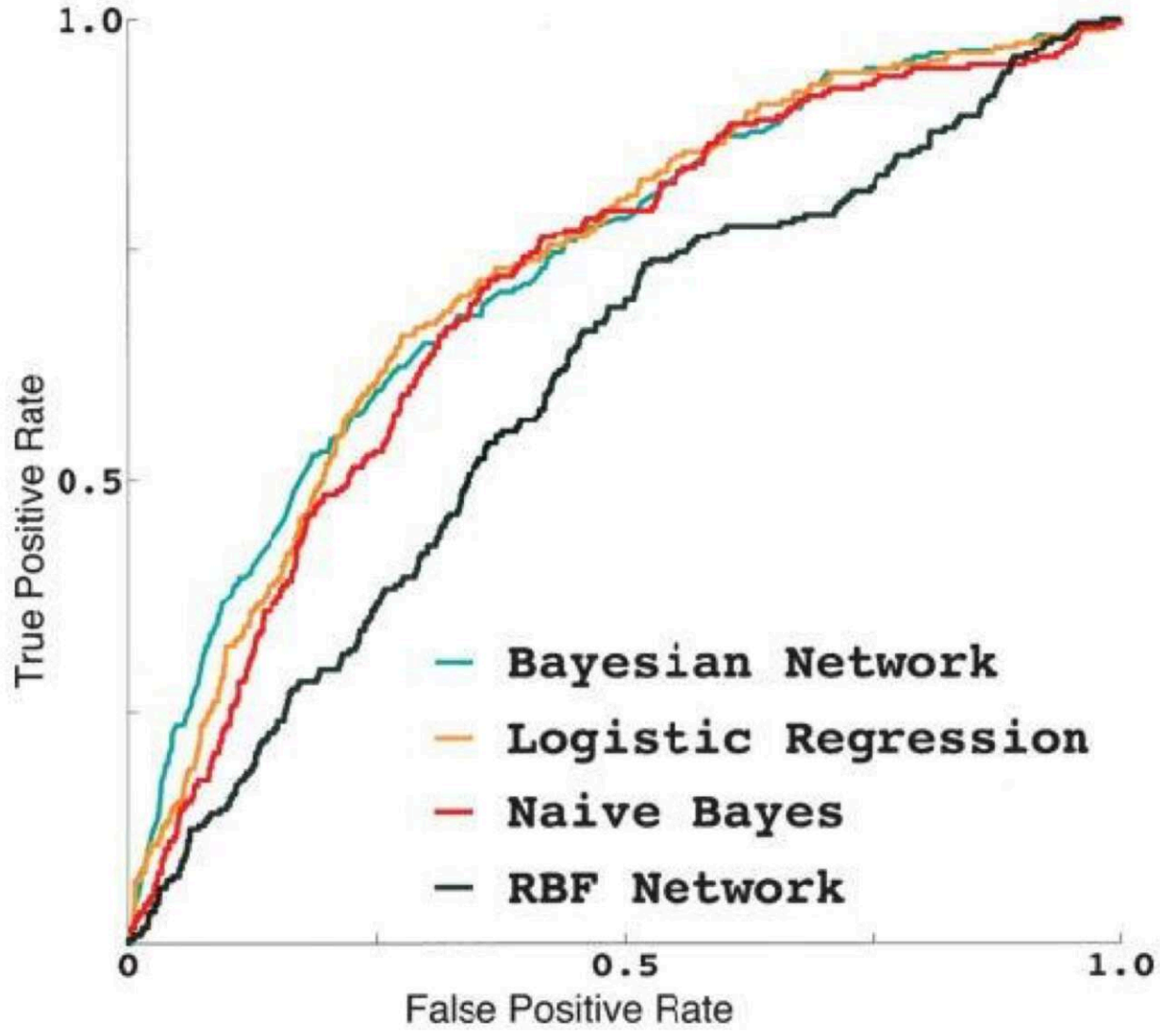
- The perfect classifier (pink line) would be a curve that reaches the northwest corner
- This represents a zero false positive rate and a 100% true positive rate

- A metric related to the ROC curve is the **area under the curve (AUC)**
- Notice that for the perfect classifier, the AUC would equal 1
- An AUC closer to 1 is better, and it ranges from 0 to 1



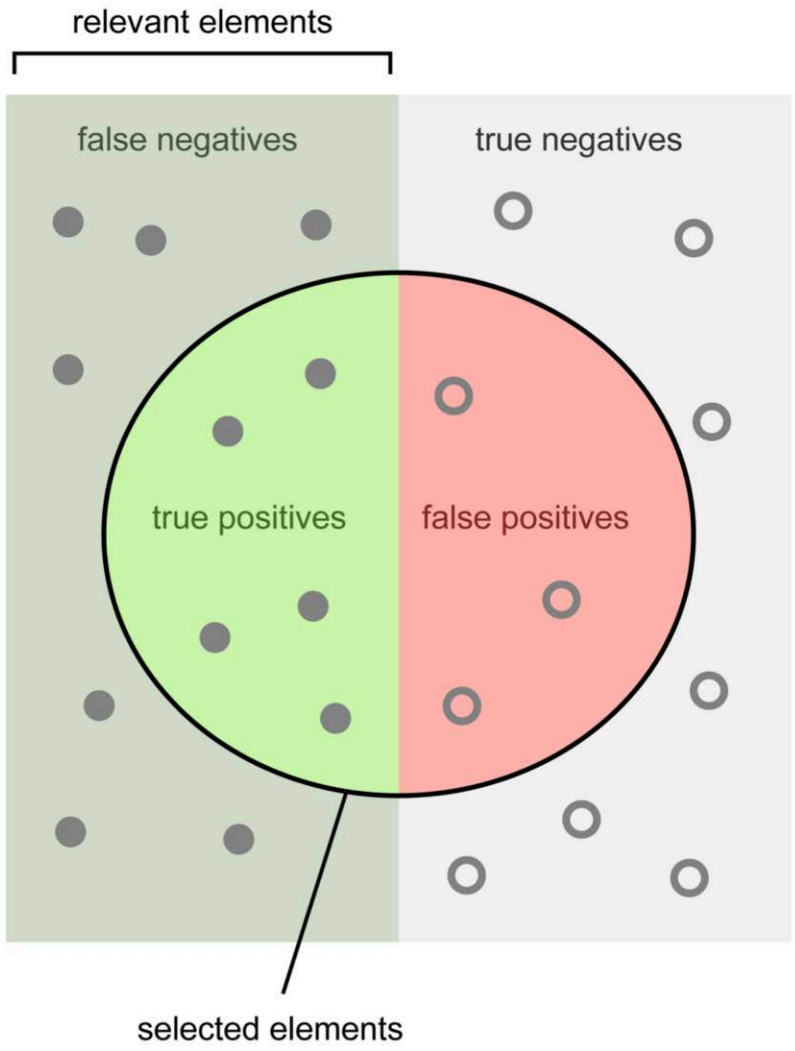
Images from Wikipedia.

Using a ROC curve to compare algorithms



Accuracy-based metrics: Precision and recall

When getting one class correct is more important



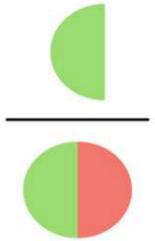
We don't care about the accuracy of all classes equally

E.g. If we're trying to detect credit card fraud or the presence of a rare disease

Sometimes we're willing to trade-off misclassifying one class to get better accuracy in a different class

Let's define precision and recall:

How many selected items are relevant?

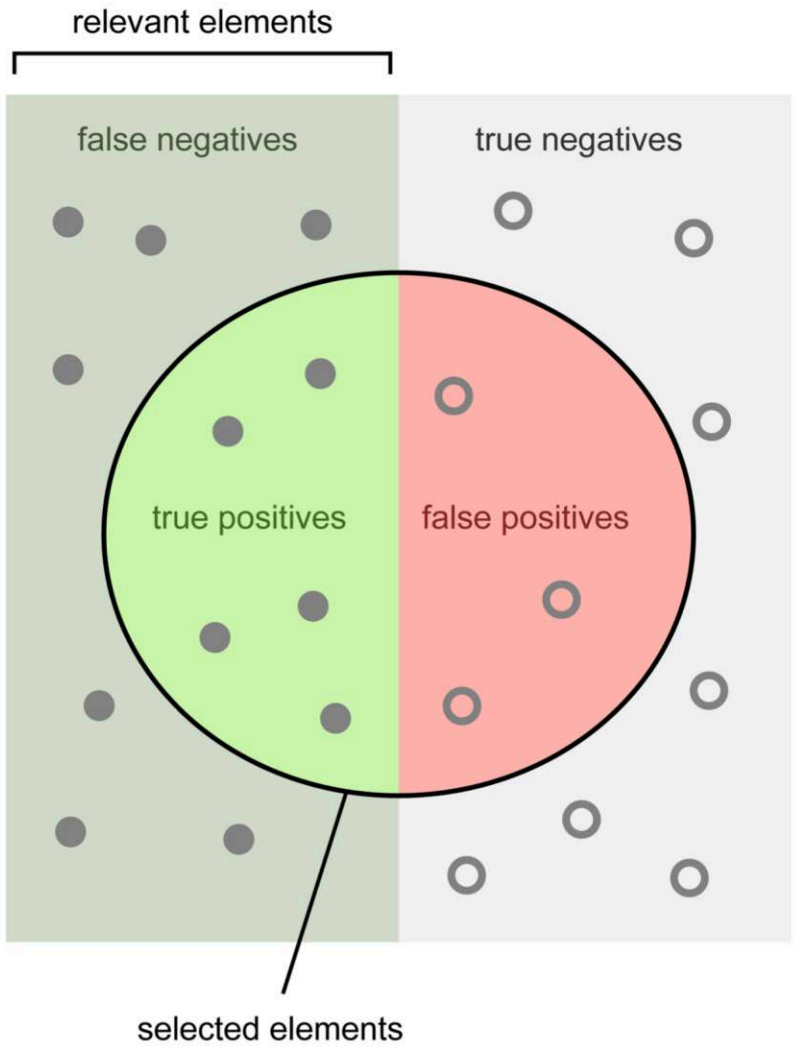
Precision = 

How many relevant items are selected?

Recall = 

Images from Wikipedia.

Precision and recall



How many selected items are relevant?

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

How many relevant items are selected?

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

Student exercise:

- Can you name cases where we may care more about precision?
- What about cases where we care more about recall?

Images from Wikipedia.

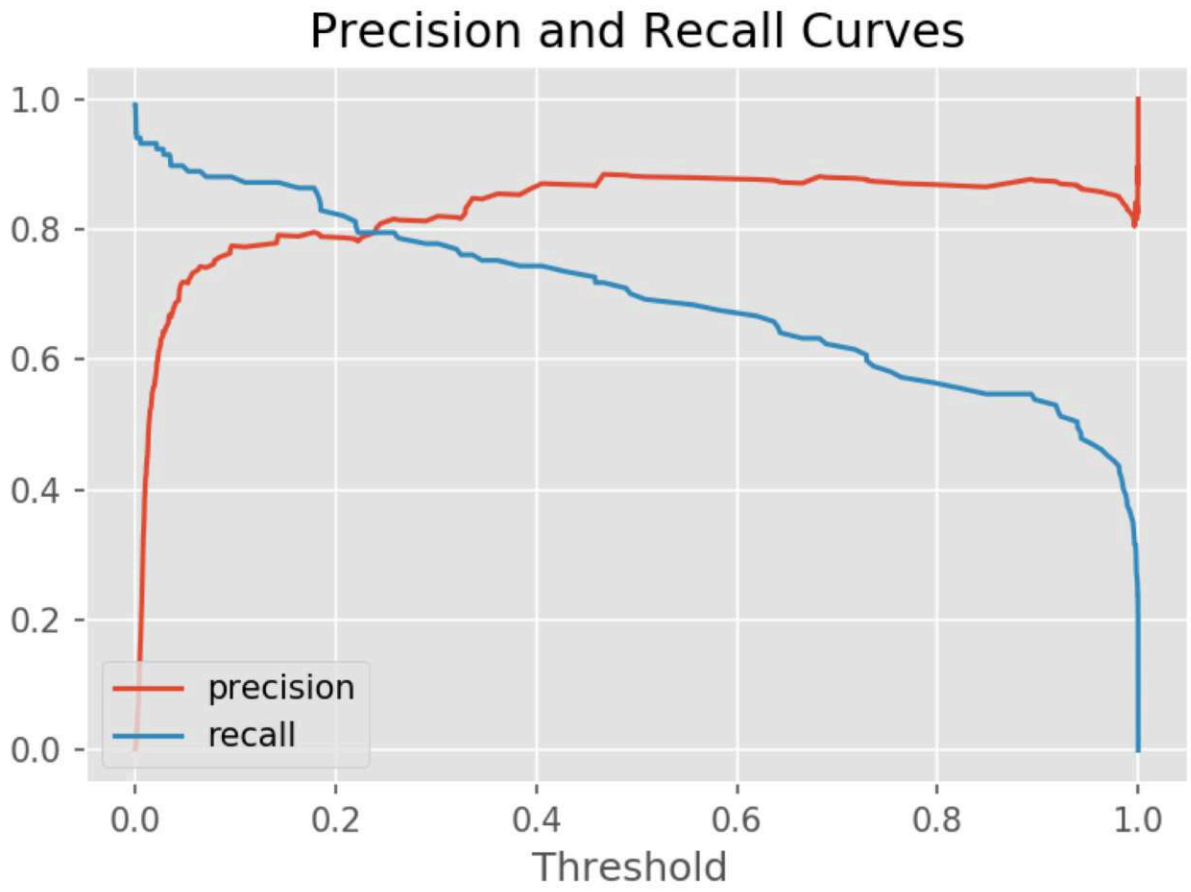
The precision-recall curve

We saw earlier that we can change our threshold, and so change the classifications of our model

As we move the threshold, we will also change our precision and recall

Want to find every positive class in the data? Then decrease the threshold to almost nothing: viola, almost all observations will be classified as positive

Want to make sure what you classify as positive is truly positive? Increase the threshold and make it harder for the model to classify observations as positive



Even more classification metrics

More classification metrics

		Condition (as determined by "Gold standard")		
	Total population	Condition positive	Condition negative	Prevalence = Σ Condition positive / Σ Total population
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV, Precision) = Σ True positive / Σ Test outcome positive
	Test outcome negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) = Σ False negative / Σ Test outcome negative
	Positive likelihood ratio (**LR+**) = TPR/FPR	True positive rate (TPR, Sensitivity, Recall) = Σ True positive / Σ Condition positive	False positive rate (FPR, Fall-out) = Σ False positive / Σ Condition negative	Accuracy (ACC) = Σ True positive + Σ True negative / Σ Total population
	Negative likelihood ratio (**LR−**) = FNR/TNR	False negative rate (FNR) = Σ False negative / Σ Condition positive	True negative rate (TNR, Specificity, SPC) = Σ True negative / Σ Condition negative	
	Diagnostic odds ratio (**DOR**) = LR+/LR−			

	False discovery rate (FDR) = Σ False positive / Σ Test outcome positive
	Negative predictive value (NPV) = Σ True negative / Σ Test outcome negative

Image from Wikipedia.

More classification metrics

		Condition (as determined by "Gold standard")			
	Total population	Condition positive	Condition negative	Prevalence = Σ Condition positive / Σ Total population	
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV, Precision) = Σ True positive / Σ Test outcome positive	False discovery rate (FDR) = Σ False positive / Σ Test outcome positive
	Test outcome negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) = Σ False negative / Σ Test outcome negative	Negative predictive value (NPV) = Σ True negative / Σ Test outcome negative
	Positive likelihood ratio (**LR+**) = TPR/FPR	True positive rate (TPR, Sensitivity, Recall) = Σ True positive / Σ Condition positive	False positive rate (FPR, Fall-out) = Σ False positive / Σ Condition negative	Accuracy (ACC) = Σ True positive + Σ True negative / Σ Total population	
	Negative likelihood ratio (**LR−**) = FNR/TNR	False negative rate (FNR) = Σ False negative / Σ Condition positive	True negative rate (TNR, Specificity, SPC) = Σ True negative / Σ Condition negative		
	Diagnostic odds ratio (**DOR**) = LR+/LR−				

Image from Wikipedia.

More classification metrics

		Condition (as determined by "Gold standard")		
	Total population	Condition positive	Condition negative	Prevalence = Σ Condition positive / Σ Total population
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV, Precision) = Σ True positive / Σ Test outcome positive
	Test outcome negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) = Σ False negative / Σ Test outcome negative
	Positive likelihood ratio (**LR+**) = TPR/FPR	True positive rate (TPR, Sensitivity, Recall) = Σ True positive / Σ Condition positive	False positive rate (FPR, Fall-out) = Σ False positive / Σ Condition negative	Accuracy (ACC) = Σ True positive + Σ True negative / Σ Total population
	Negative likelihood ratio (**LR−**) = FNR/TNR	False negative rate (FNR) = Σ False negative / Σ Condition positive	True negative rate (TNR, Specificity, SPC) = Σ True negative / Σ Condition negative	
	Diagnostic odds ratio (**DOR**) = LR+/LR−			

	False discovery rate (FDR) = Σ False positive / Σ Test outcome positive
	Negative predictive value (NPV) = Σ True negative / Σ Test outcome negative

Image from Wikipedia.

More classification metrics

		Condition (as determined by "Gold standard")			
	Total population	Condition positive	Condition negative	Prevalence = Σ Condition positive / Σ Total population	
Test outcome	Test outcome positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV, Precision) = Σ True positive / Σ Test outcome positive	False discovery rate (FDR) = Σ False positive / Σ Test outcome positive
	Test outcome negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) = Σ False negative / Σ Test outcome negative	Negative predictive value (NPV) = Σ True negative / Σ Test outcome negative
	Positive likelihood ratio (**LR+**) = TPR/FPR	True positive rate (TPR, Sensitivity, Recall) = Σ True positive / Σ Condition positive	False positive rate (FPR, Fall-out) = Σ False positive / Σ Condition negative	Accuracy (ACC) = Σ True positive + Σ True negative / Σ Total population	
	Negative likelihood ratio (**LR−**) = FNR/TNR	False negative rate (FNR) = Σ False negative / Σ Condition positive	True negative rate (TNR, Specificity, SPC) = Σ True negative / Σ Condition negative		
	Diagnostic odds ratio (**DOR**) = LR+/LR−				

Image from Wikipedia.

Applications to model development

Fit to training, evaluate on test (or cross-val)

Reminder from our cross-validation lecture:

- Fit your model to your training dataset
- Evaluate its performance (error metrics, e.g. confusion matrix, F1 score, ROC curve) on the test dataset
- Even better: evaluate the model using k-fold cross-validation

The final error metrics you report should be from your holdout dataset

- Fit on training dataset
- Keep evaluating on the test dataset while doing model development
- **Report the error metrics using a model you fit on training, but evaluated on the holdout**

Questions? 🤔