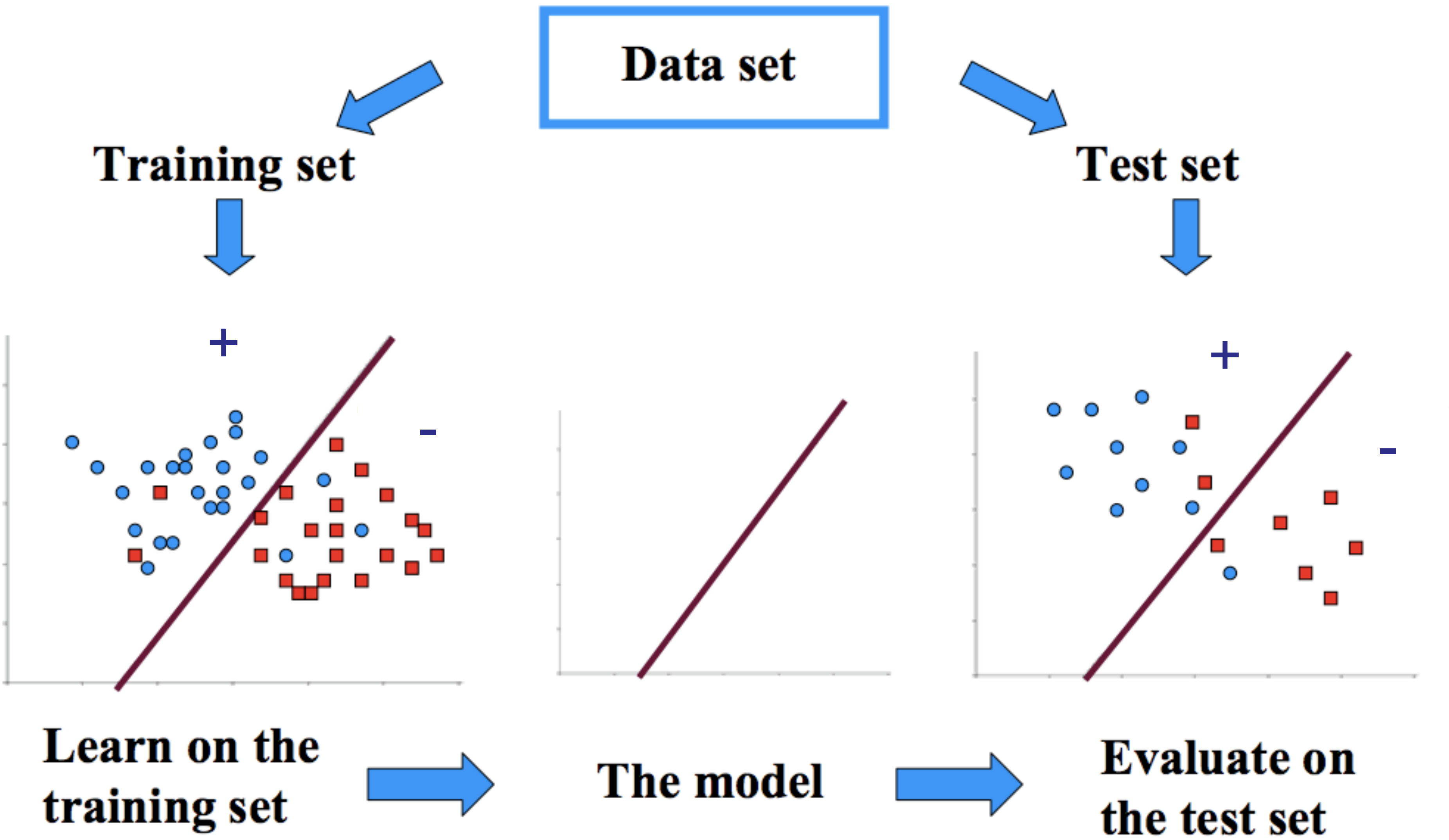
Practical Issues

June 20, 2016

Credits for slides: Allan, Arms, Manning, Lund, Noble, Page.

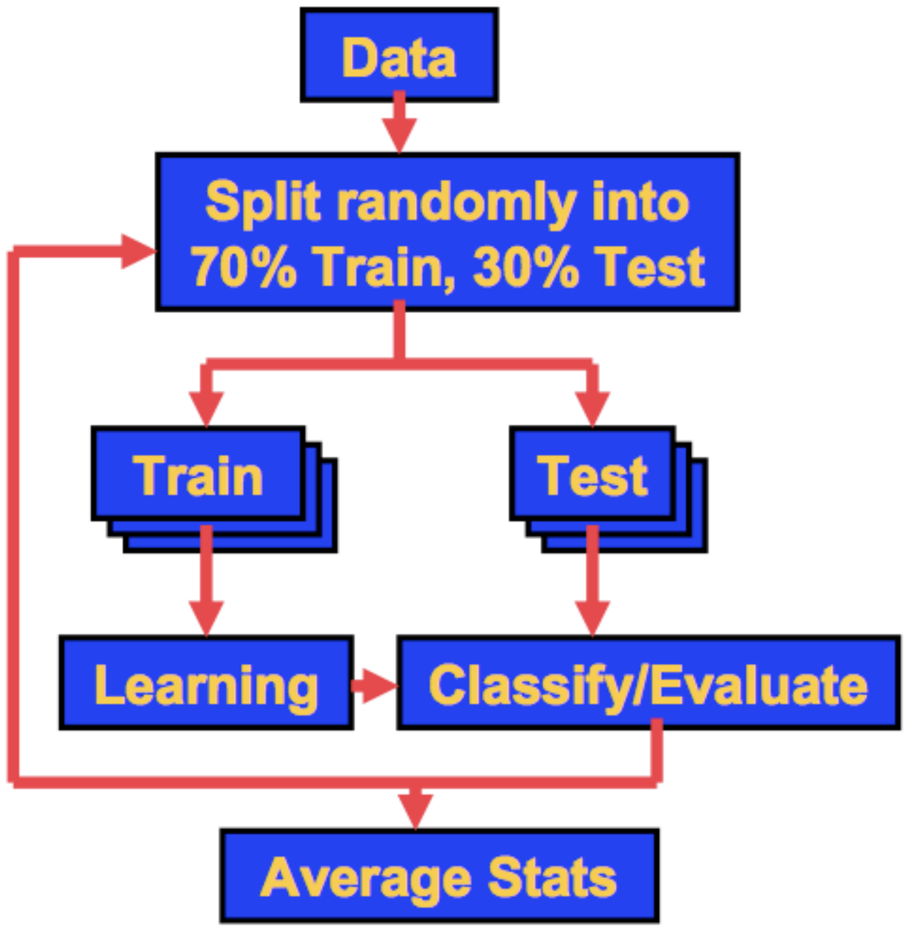
Evaluation Framework



Methods to Estimate Performance

- Holdout
  - Reserve ½ for training and ½ for testing
  - Reserve 2/3 for training and 1/3 for testing
- To limit the effect of one lucky or unlucky train/test split it is common to average through:
  - Random subsampling
    - Repeated holdout
  - Stratified sampling
  - Cross validation
    - Partition data into k disjoint subsets
    - k-fold: train on k-1 partitions, test on the remaining one
    - Leave-one-out: k=n

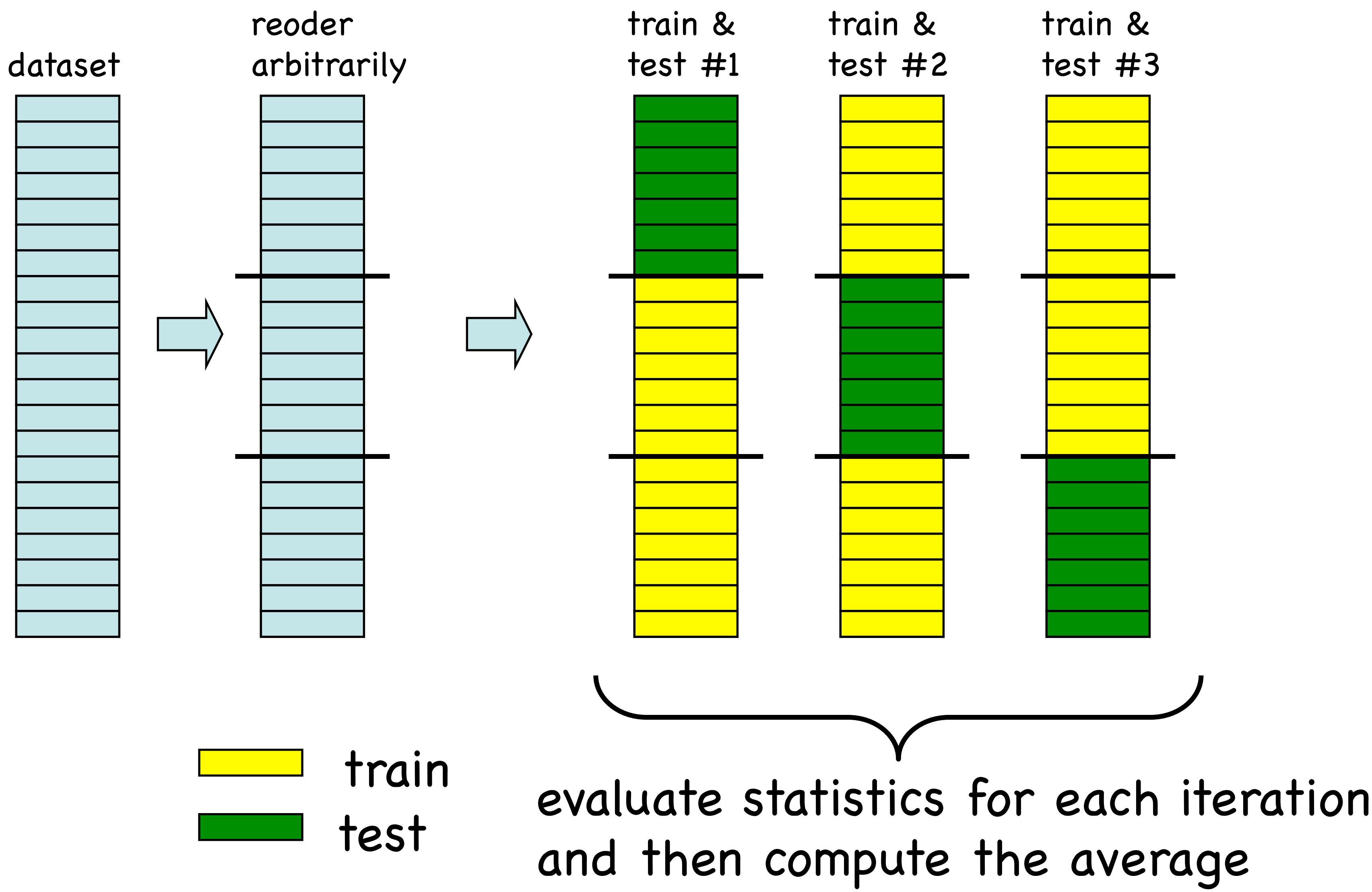
Random sub-sampling



Stratified Sampling

- The holdout method reserves a certain amount for testing and uses the remainder for training
- For small or “unbalanced” datasets, training samples might not be representative for all classes
- For instance, only few instances of some classes
- Stratified sample
  - Make sure that each class is represented with approximately equal proportions in both subsets

3-Fold Cross Validation



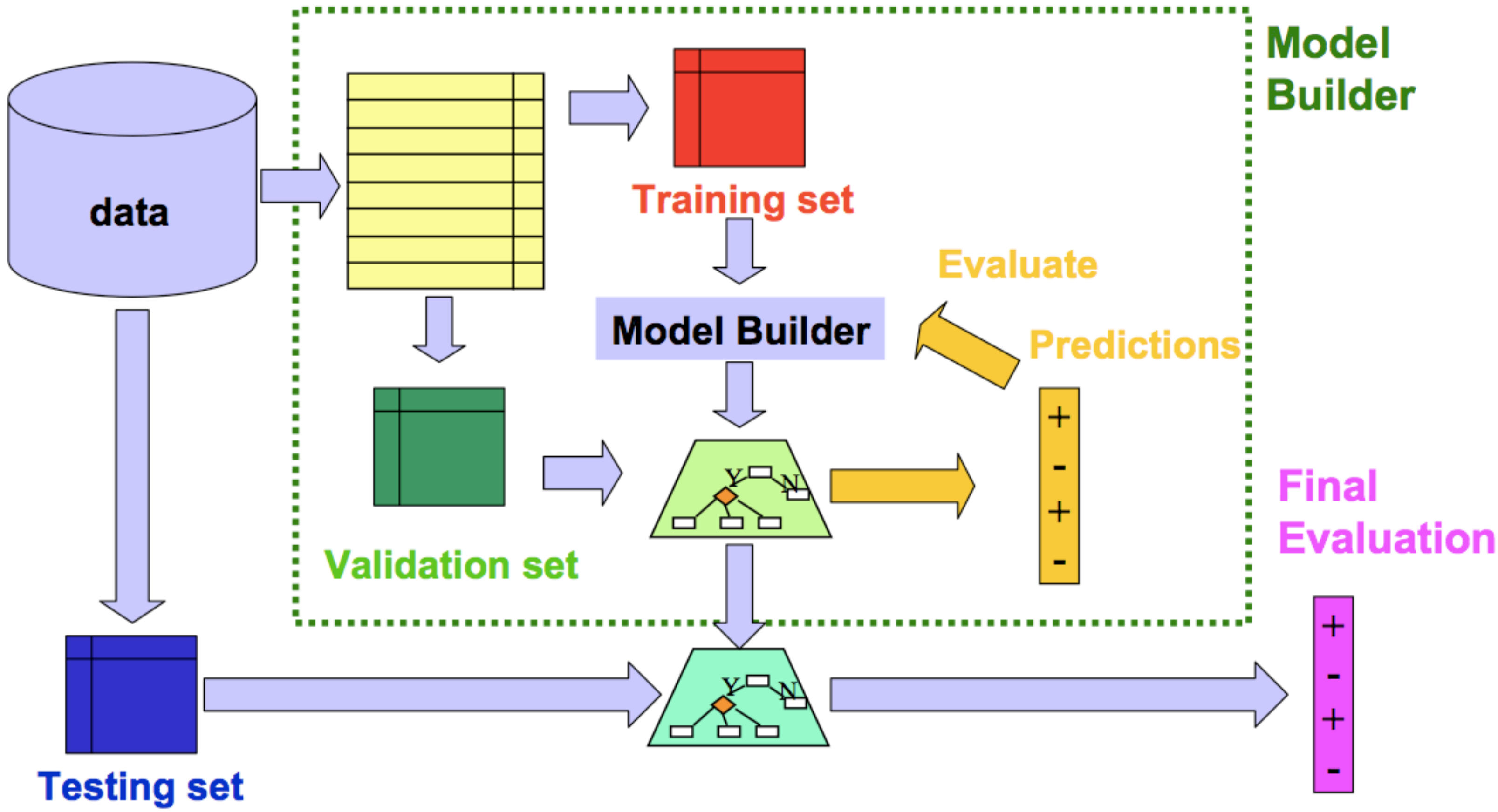
k-Fold Cross Validation

- Split the data to k sets of approximately equal size (and class distribution, if stratified)
- For i=1 to k:
  - Use i-th subset for testing and remaining (k-1) subsets for training
- Compute average accuracy
- k-fold CV can be repeated several, say, 10 times

A Note on Parameter Tuning

- It is important that the test data is not used in any way to create the classifier
- Some learning schemes operate in two stages:
  - Stage 1: builds the basic structure
  - Stage 2: optimizes parameter settings
- The test data can’t be used for parameter tuning!
- Proper procedure uses three sets:
  - training data, validation data, and test data
- Validation data is used to optimize parameters

Train, Validation, and Test



Test Statistics:
Contingency Table of Classification Results

		True Class		
		+	-	Totals
Result from classification model	+	TP	FP	TP+FP
	-	FN	TN	FN+TN
	Totals	TP+FN	FP+TN	N

true positive, false positive
false negative, true negative

Classification Accuracy

		True Class		
		+	-	Totals
Result from classification model	+	TP	FP	TP+FP
	-	FN	TN	FN+TN
	Totals	TP+FN	FP+TN	N

- CA = (TP+TN) / N
- Proportion of correctly classified examples

Sensitivity

		True Class		
		+	-	Totals
Result from classification model	+	TP	FP	TP+FP
	-	FN	TN	FN+TN
	Totals	TP+FN	FP+TN	N

- Sensitivity = TP / (TP + FN)
- Proportion of correctly detected positive examples
- In medicine (+, -: presence and absence of a disease):
  - chance that our model correctly identifies a patient with a disease

Specificity

True Class

		+	-	Totals
Result from classification model	+	TP	FP	TP+FP
	-	FN	TN	FN+TN
	Totals	TP+FN	FP+TN	N

- Specificity = TN / (FP + TN)
- Proportion of correctly detected negative examples
- In medicine:
  - chance that our model correctly identifies a patient without a disease

To summarize: Evaluation Measures

Start with a CONTINGENCY table

	actual +	actual -
predicted +	*TP*	*FP*
predicted -	*FN*	*TN*

Of all patients that actually have the disease, what fraction did we correctly detect as having the disease?

Sensitivity/Recall

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

Precision

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

Of all patients we predicted +, what fraction actually have the disease?

Accuracy $= \dfrac{TP+TN}{N}$ Error $= \dfrac{FP+FN}{N}$

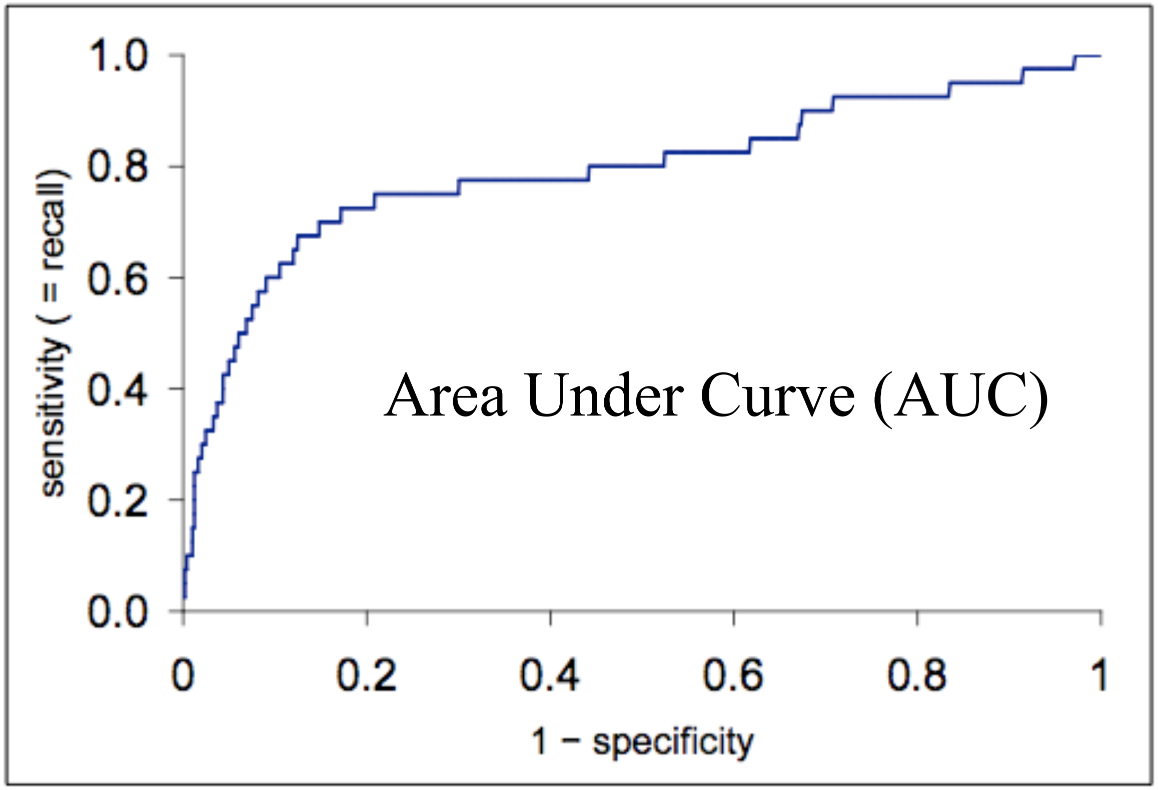
Specificity

$$SP = \frac{TN}{TN+FP}$$

where N=TP+FP+FN+TN

Receiver Operating Characteristic (ROC) Curve

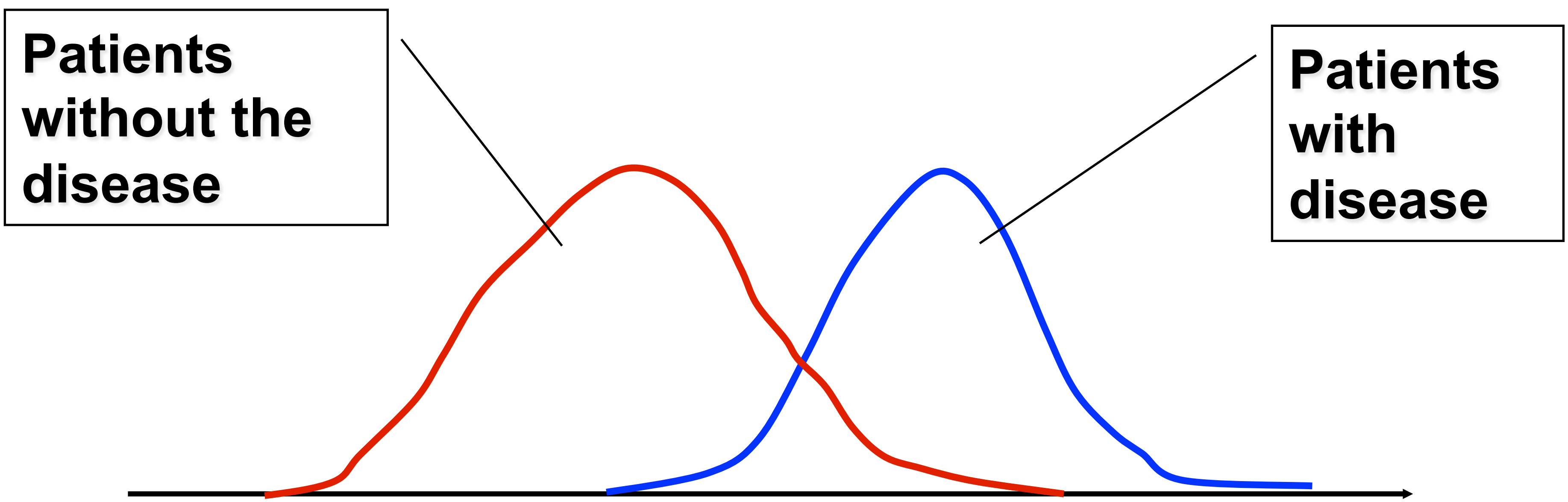
True positive rate



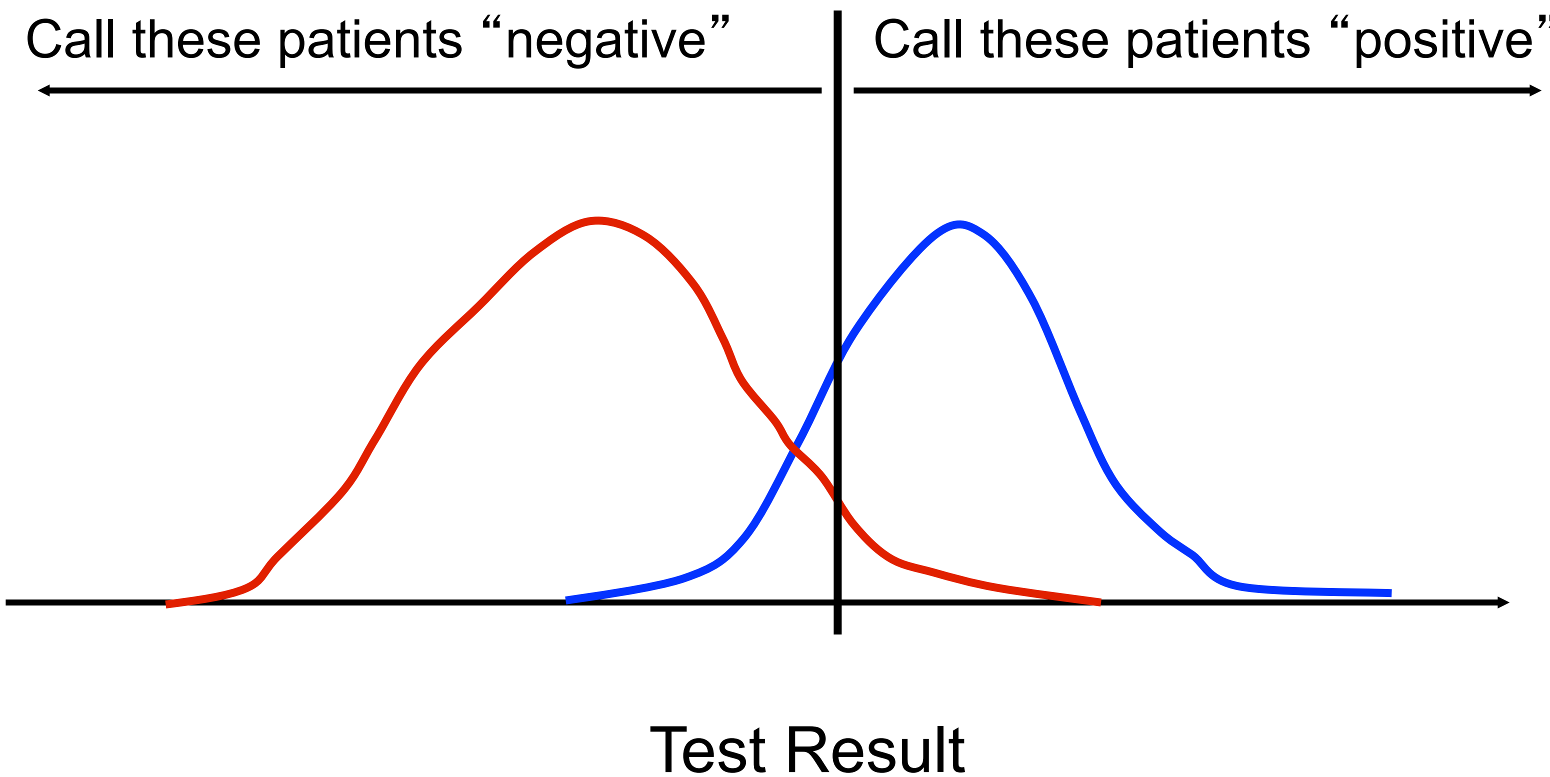
False positive rate

$$FPR = \frac{FP}{FP + TN}$$

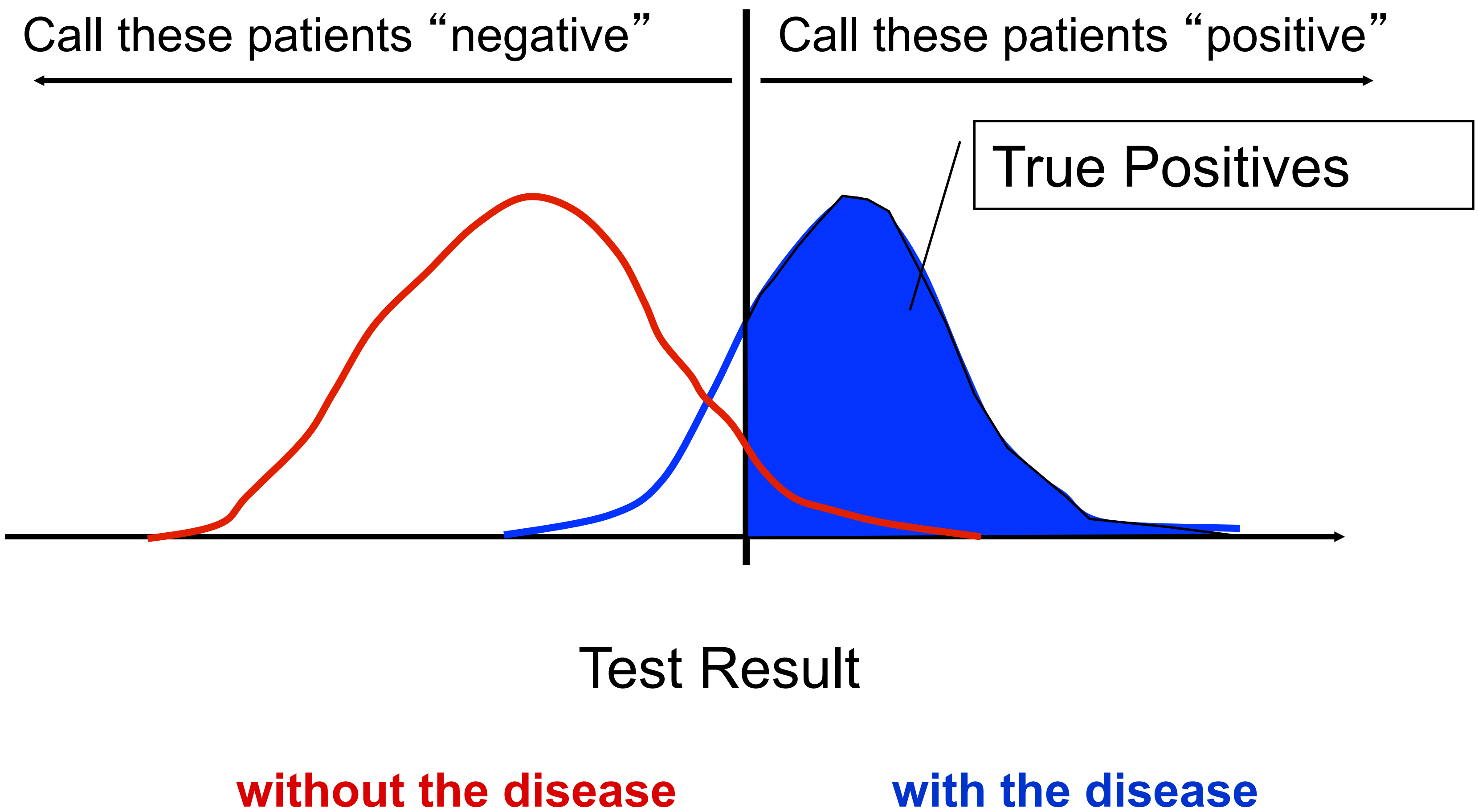
How to Draw an ROC Curve?

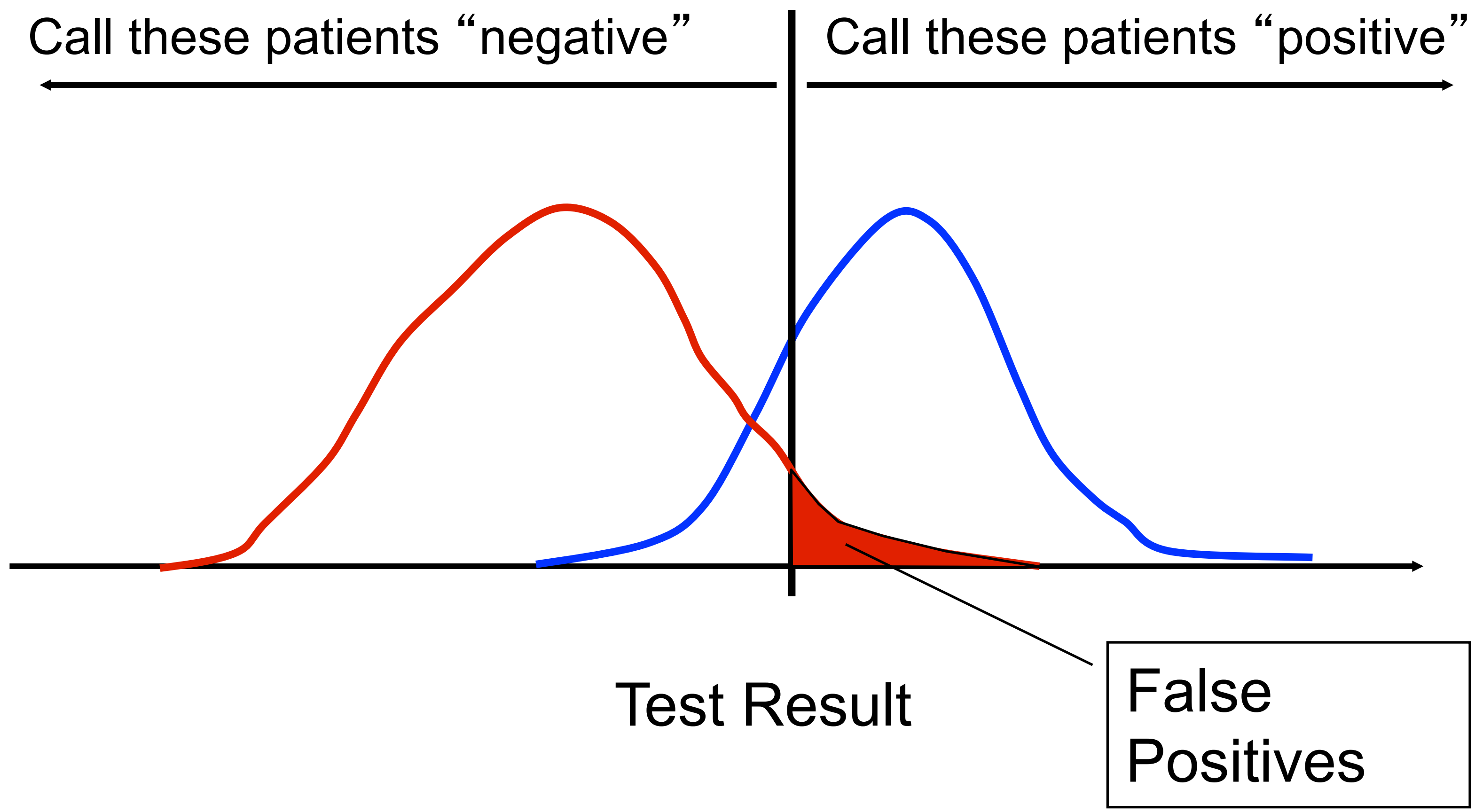


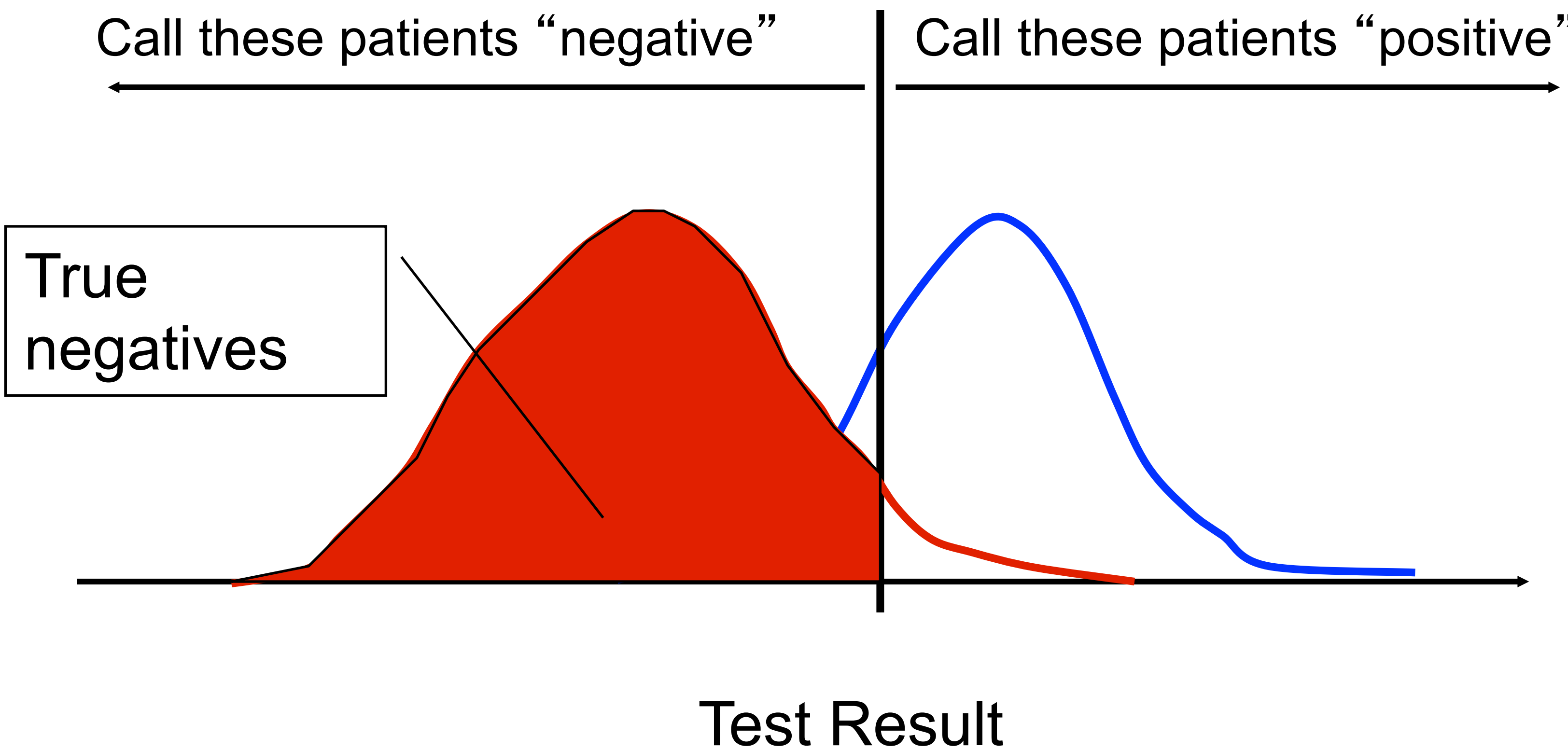
Threshold

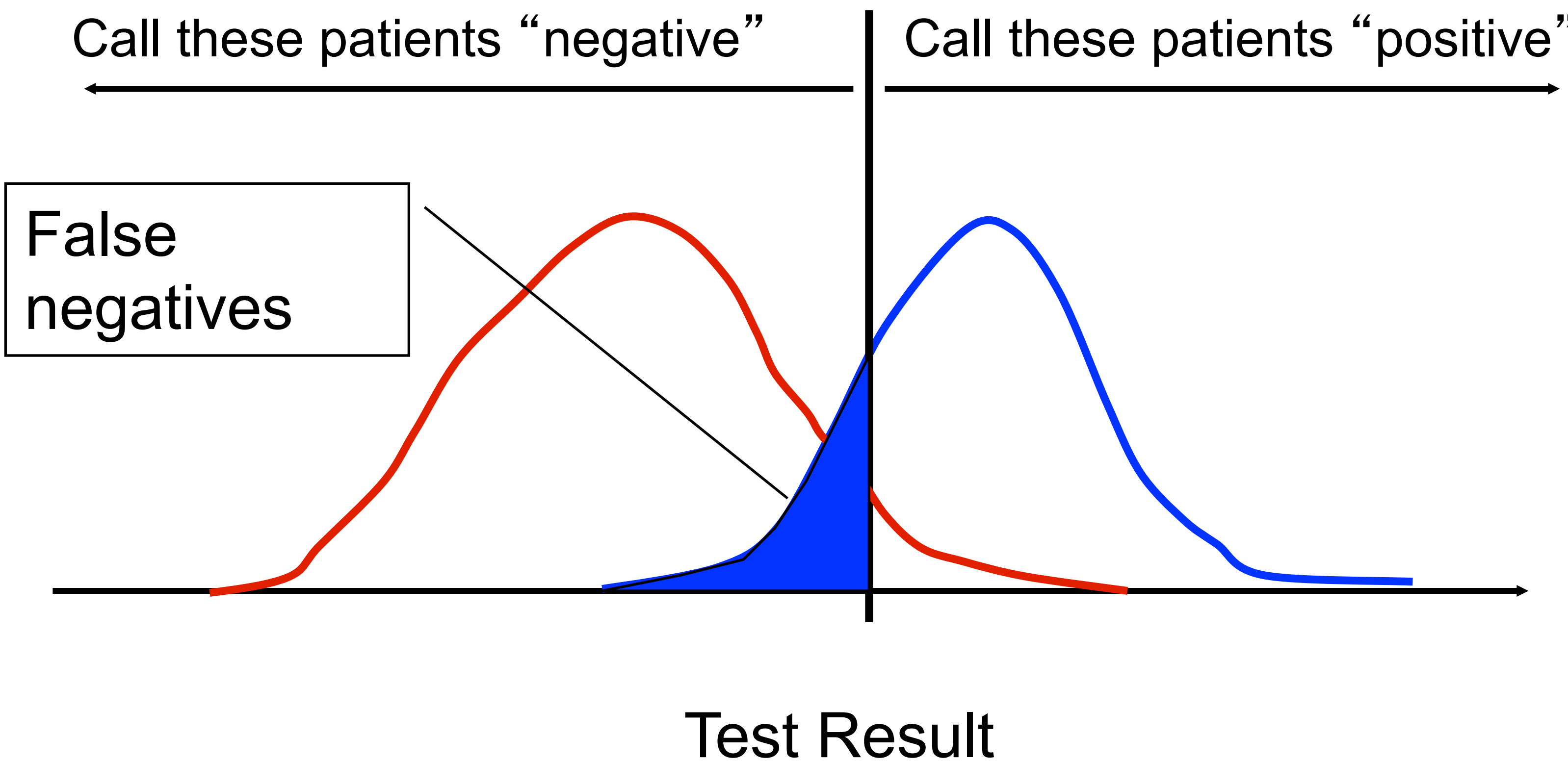


$$\frac{P(+\mid x)}{P(-\mid x)} > \theta$$

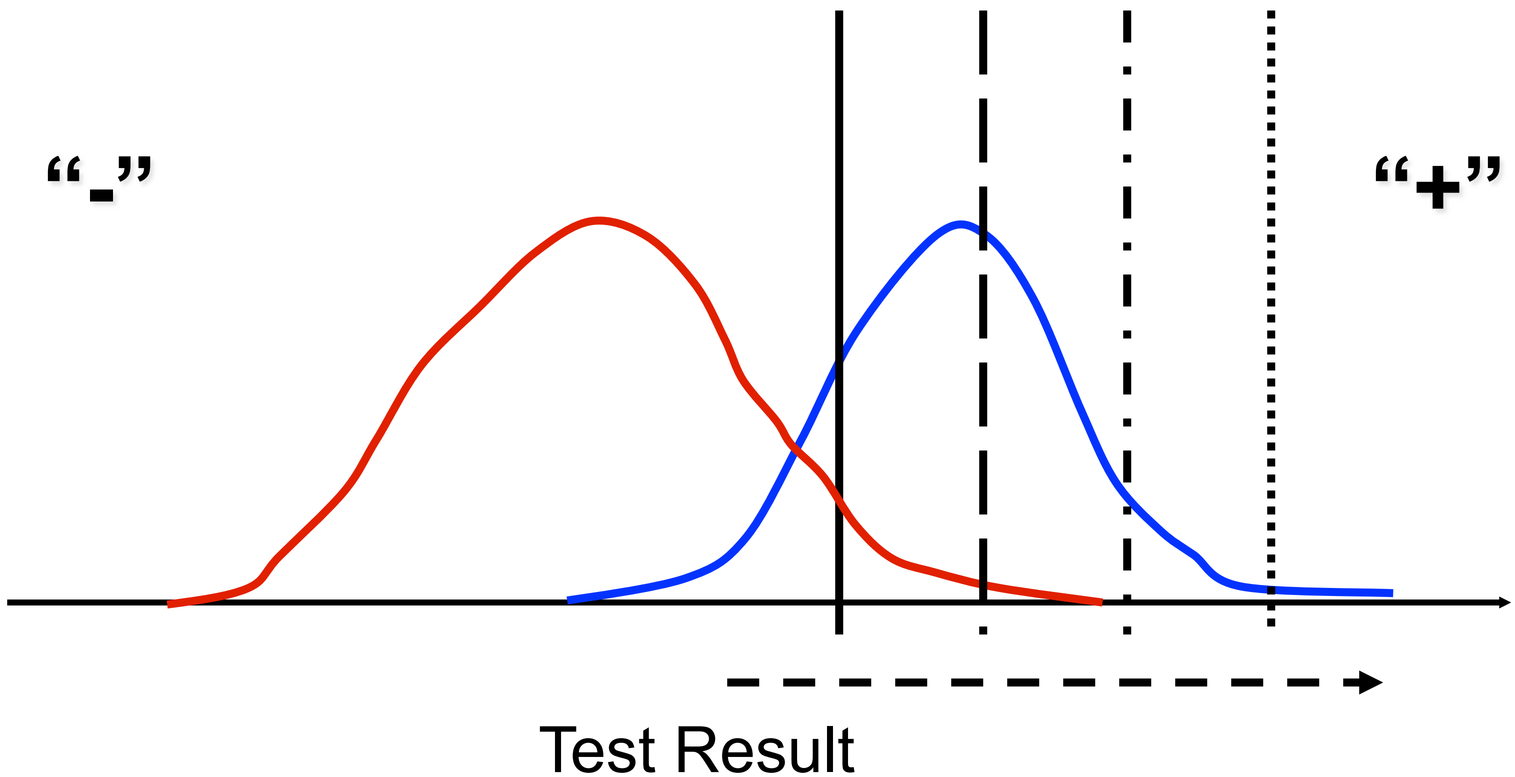

Call these patients "negative"
Call these patients "positive"
True Positives
Test Result
without the disease
with the disease


Call these patients “negative”
Call these patients “positive”
Test Result
False Positives

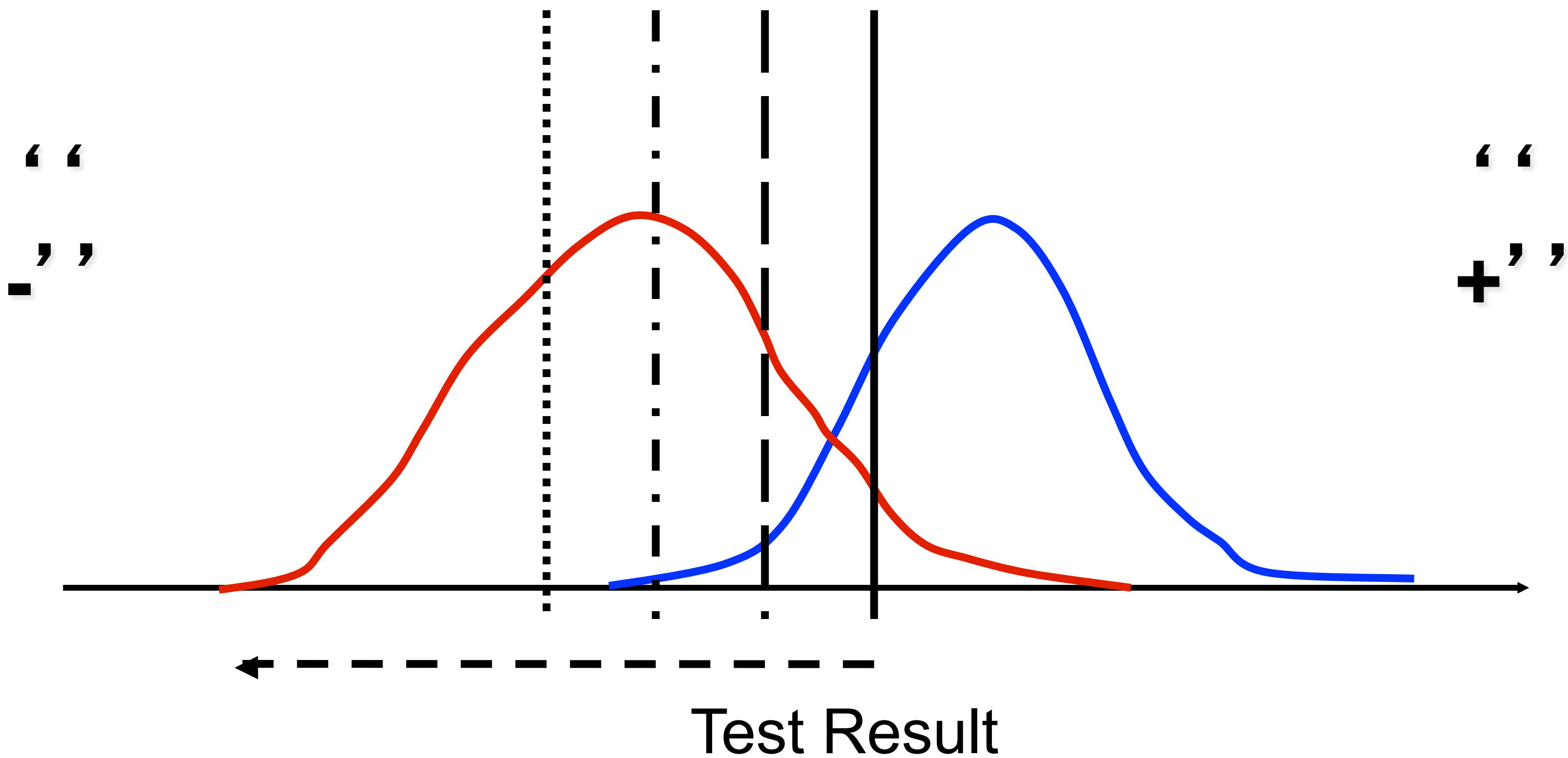

Call these patients "negative"
Call these patients "positive"
True negatives
Test Result


Call these patients "negative"
Call these patients "positive"
False negatives
Test Result

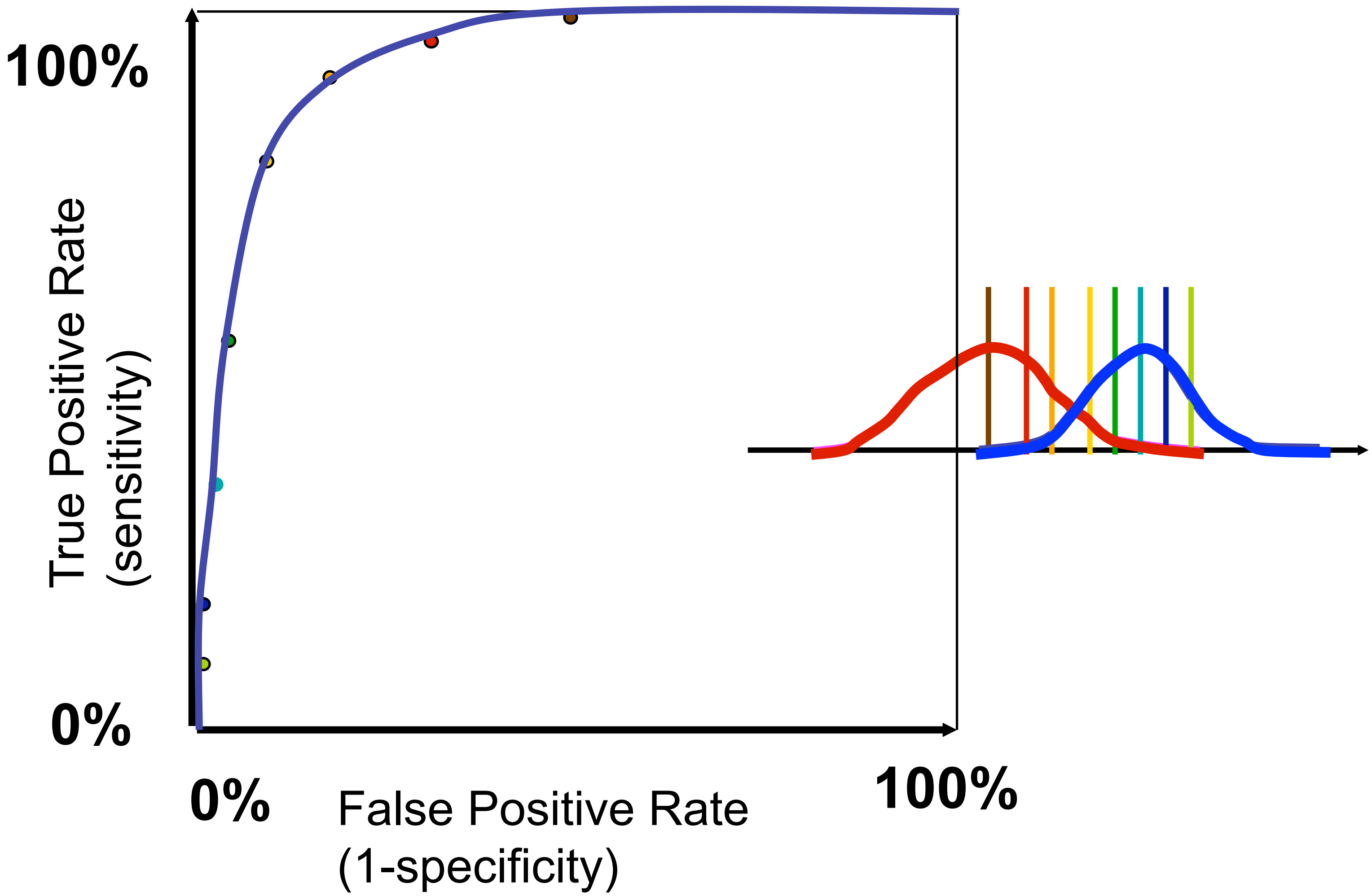
Moving the Threshold: right



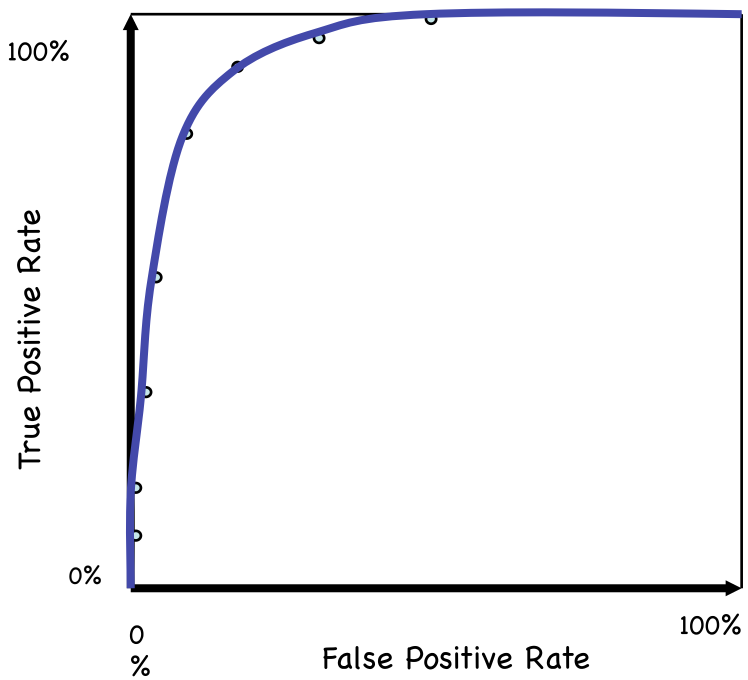
Moving the Threshold: left

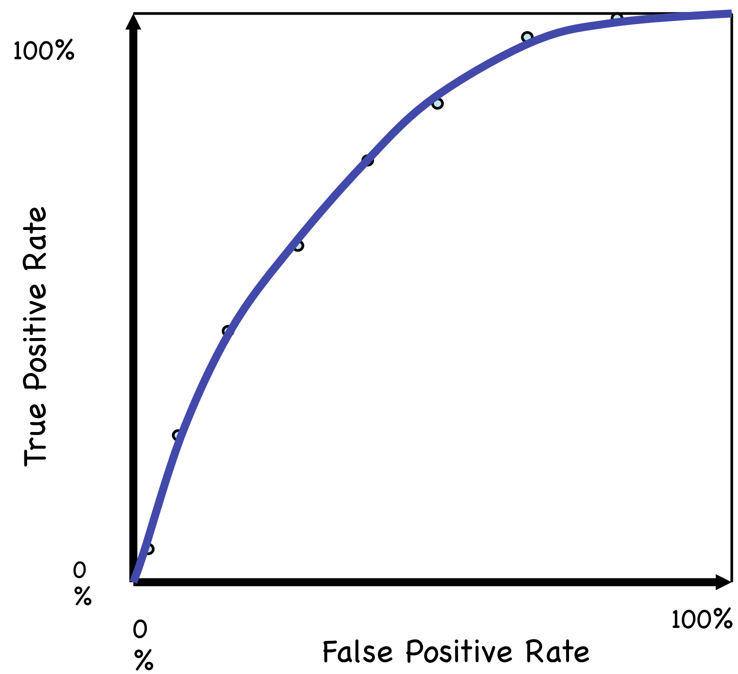


Receiver Operating Characteristic (ROC) Curve

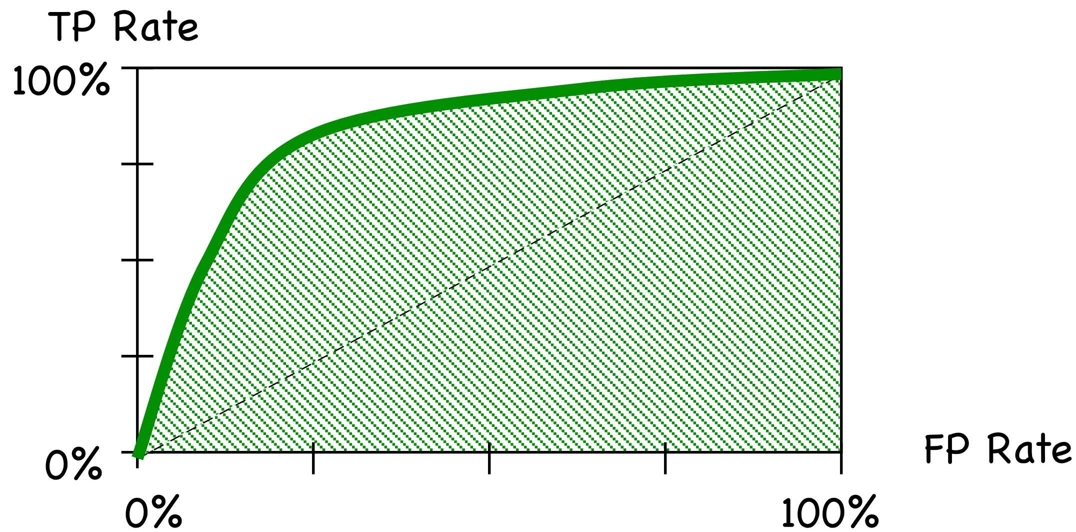


ROC Curve Comparison





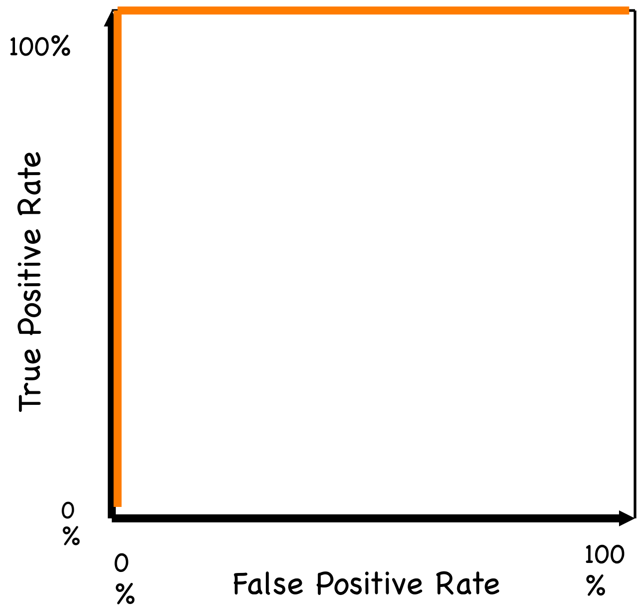
Area Under ROC



- Is expected to be from 0.5 to 1.0
- The score is not affected by class distributions
- Characteristic landmarks
  - 0.5: random classifier
  - below 0.7: poor classification
  - 0.7 to 0.8: ok, reasonable classification
  - 0.8 to 0.9: here is where very good predictive models start

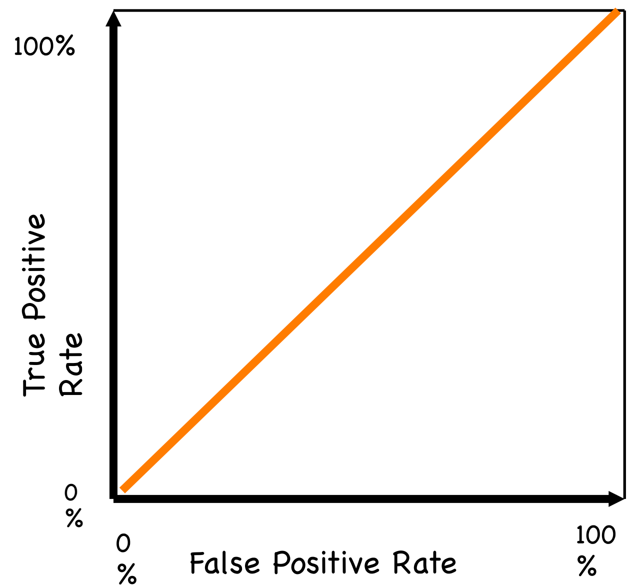
ROC Curve Extremes

Best classifier:



AUC = 1 perfect discrimination

Worst classifier:



AUC = 0.5 random discrimination

AUC = probability of correct discrimination

Comparing Two Learning Schemes

- Frequent question: which of two learning schemes performs better?
- Note: this is domain dependent!
- Obvious way: compare 10-fold CV estimates
- Generally sufficient in applications (we don't lose if the chosen method is not truly better)
- However, what about machine learning research?
- Need to show convincingly that a particular method works better

Final Thoughts

- Never test on the learning set
- Use some sampling procedure for testing

- Bottom line: good models are those that are useful in practice!