CS 189/289

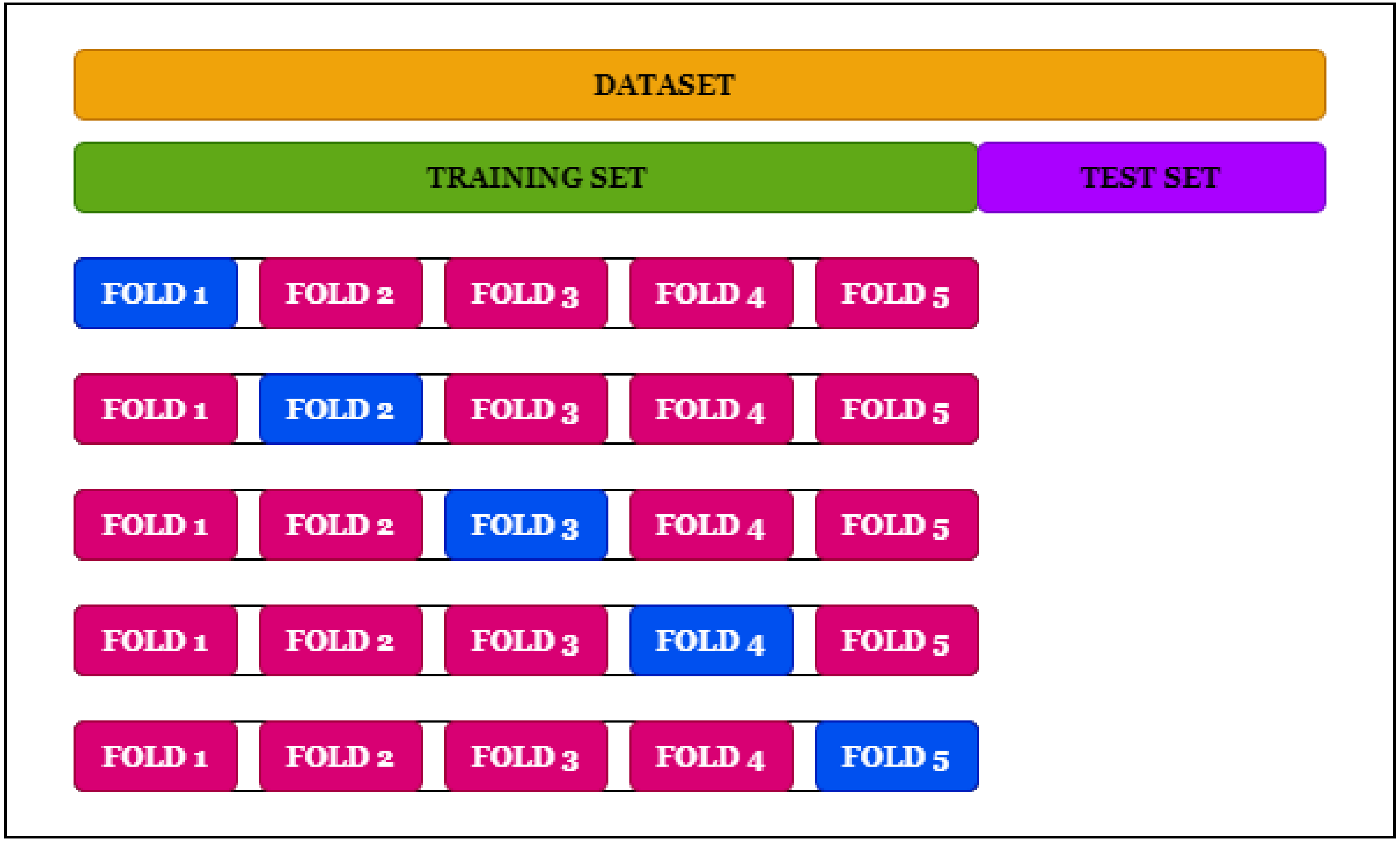
Today's lecture outline

Methods for Evaluating Classifiers

Assigned reading:

5.25, 5.26 (classifier accuracy, ROC curves)

Previously: cross-validation (CV)



- CV: re-use your data to train/validate, with one final test set.
- Suppose we were evaluating classifiers, what might we compute for each test fold?

https://medium.com/the-owl/k-fold-cross-validation-in-keras-3ec4a3a00538

How to pick between these two classifiers?

TRUTH	Logistic regression	Neural network
1	0.7198	0.9038
0	0.2460	0.8455
0	0.1219	0.4655
0	0.1560	0.3204
0	0.7527	0.2491
1	0.3064	0.7129
0	0.7194	0.4983
0	0.5531	0.6513
1	0.2173	0.3806
0	0.0839	0.1619
1	0.8429	0.7028

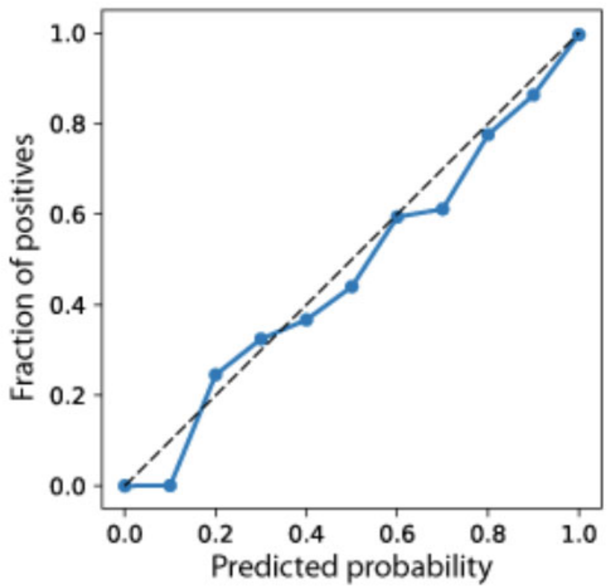
What evaluation metric/quantity applied to these data could help decide which model to use?

[Adapted https://hiplab.mc.vanderbilt.edu/people/malin/presentations/ROC_Curves.ppt]

How to pick between these two classifiers?

- We could use threshold of p=0.5 and count the # of misclassifications that each model makes ("accuracy").
- Assumes probabilities are "calibrated".
- Similarly so does hold out log likelihood.
- Suppose model is not *calibrated*, but there exists a threshold other than 0.5 that yields perfect prediction. Is this a good classifier?
- Also, what if the model is not probabilistic?
- ROC curves are going to help us deal with these issues.

TRUTH	Logistic regression	Neural network
1	0.7198	0.9038
0	0.2460	0.8455
0	0.1219	0.4655
0	0.1560	0.3204
0	0.7527	0.2491
1	0.3064	0.7129
0	0.7194	0.4983
0	0.5531	0.6513
1	0.2173	0.3806
0	0.0839	0.1619
1	0.8429	0.7028



[https://stats.stackexchange.com/questions/452483/why-some-algorithms-produce-calibrated-probabilities]

A miscalibrated but useful classifier

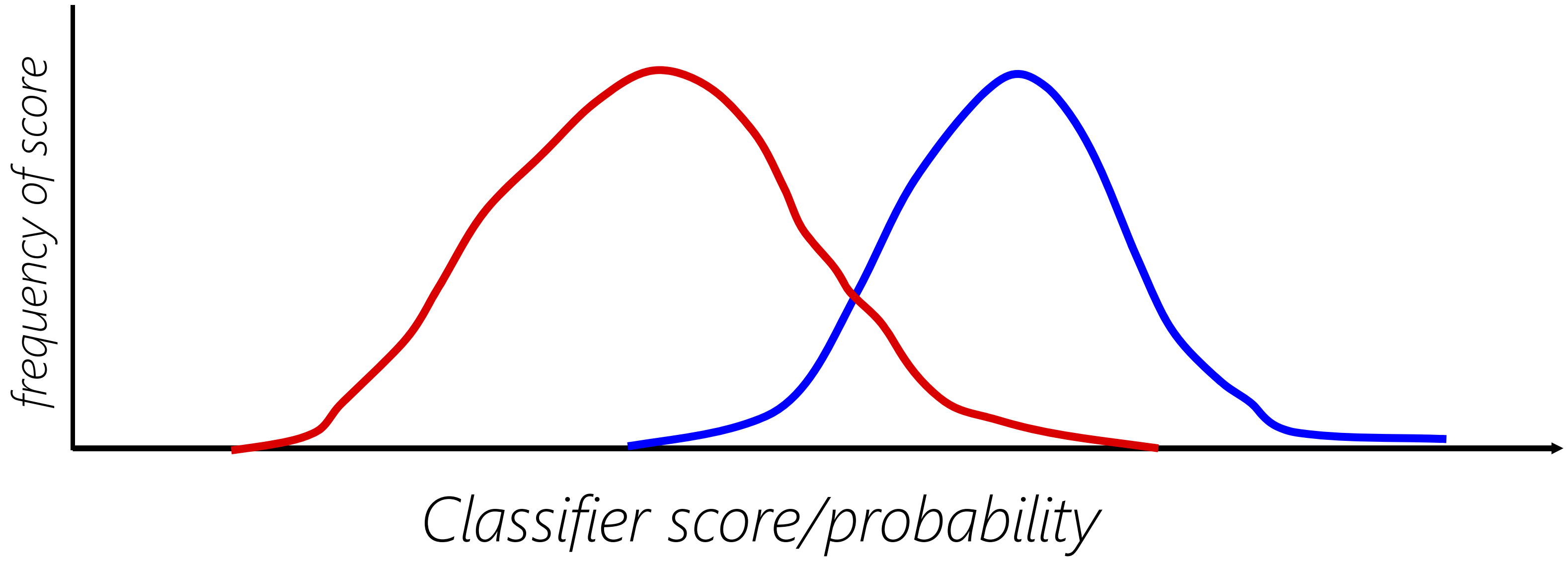
TRUTH	Logistic regression	Neural network
1	0.88	0.9038
0	0.77	0.8455
0	0.59	0.4655
0	0.81	0.3204
0	0.7527	0.2491
1	0.93	0.63
0	0.7194	0.4983
0	0.5531	0.6513
1	0.98	0.3806
0	0.0839	0.1619
1	0.8429	0.7028

- If we pick the "optimal" decision of p=0.5 decision boundary, we will choose the model "containing less information" (NN)!
- Whereas best LR threshold of 0.8 gives 0 errors, while best NN threshold of 0.5 gives 3 errors.

Defining false positives, false negatives, etc.

[We will consider only binary classifiers in today's lecture]

Distribution of classifier "scores" of unhealthy and healthy individuals in a test set



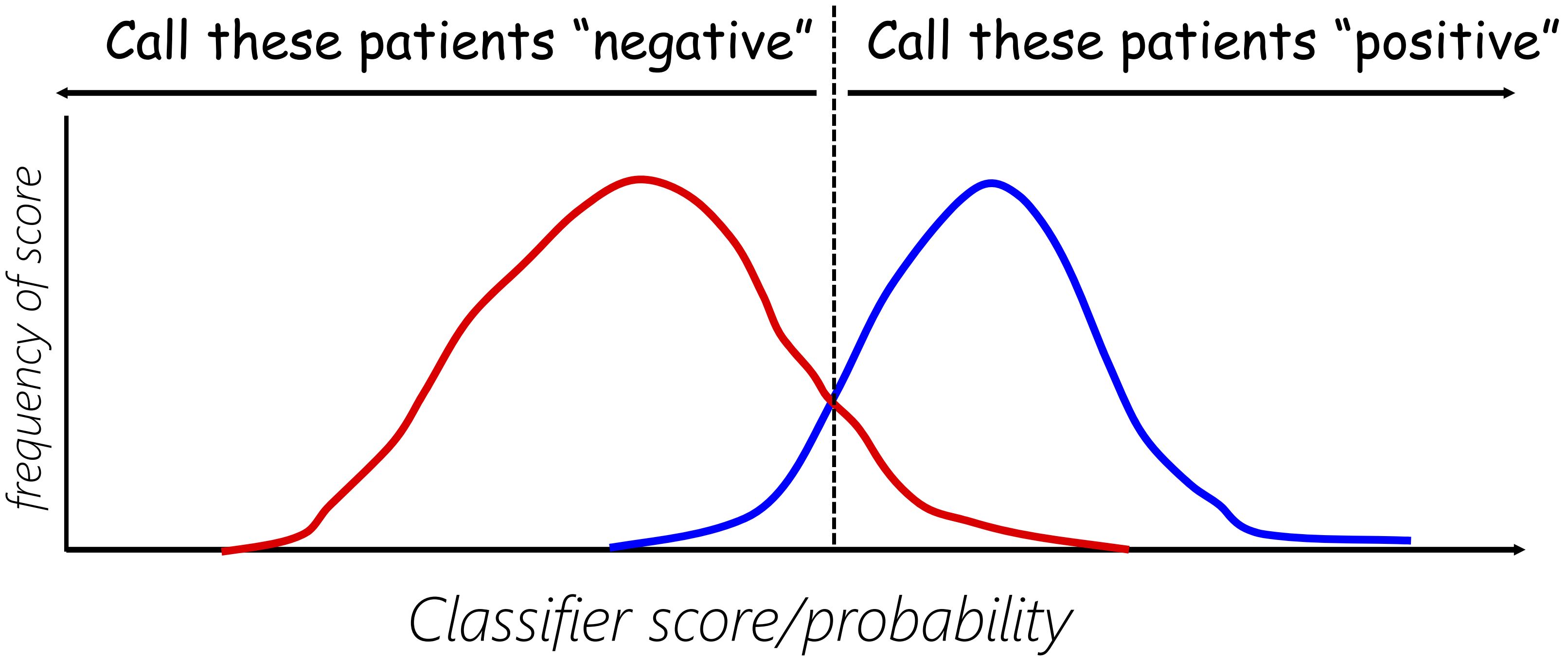
[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt]

Defining false positives, false negative, etc.

[We will consider only binary classifiers in today's lecture]

Choose a threshold on the score/probabilistic output, and call/predict all samples above it a "1" (e.g. "healthy) and all those below it a "-1" (e.g. "unhealthy").

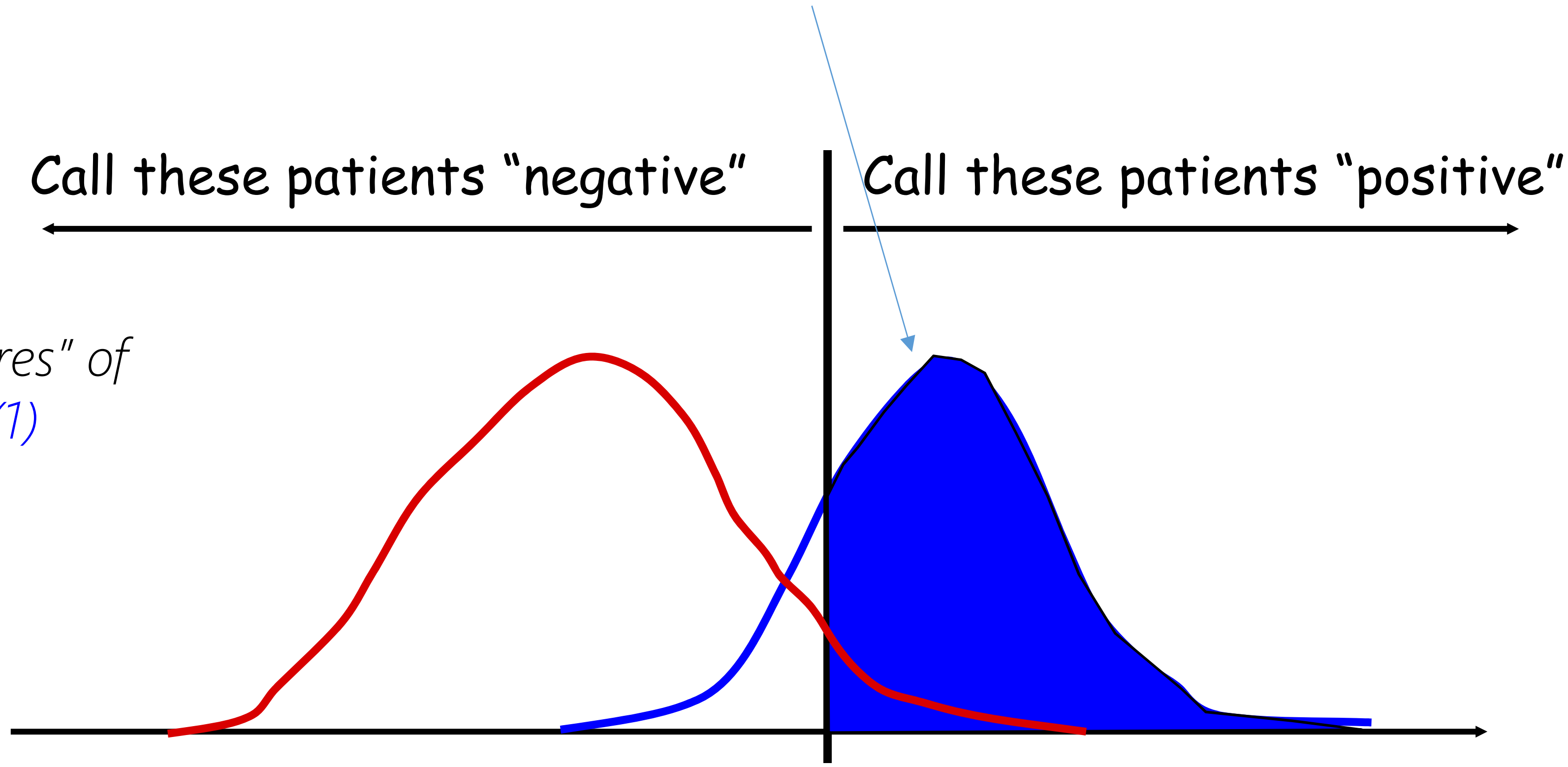
Distribution of classifier "scores" of unhealthy and healthy individuals in a test set



[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt]

Definitions: True Positives (TP)

Distribution of classifier "scores" of unhealthy (-1) and healthy (1) individuals in a test set.

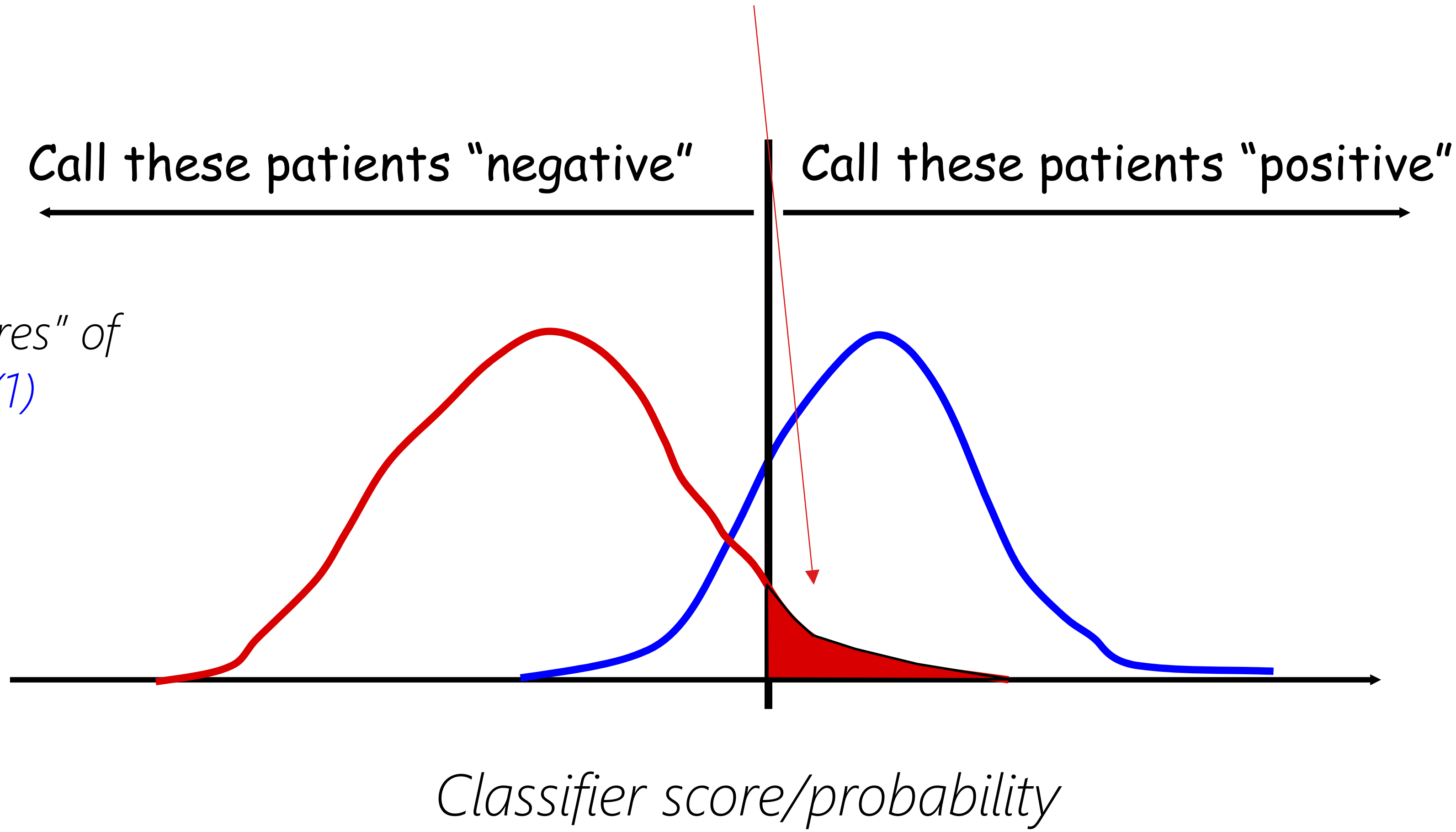


Classifier score/probability

[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt]

Definitions: False Positives (FP)

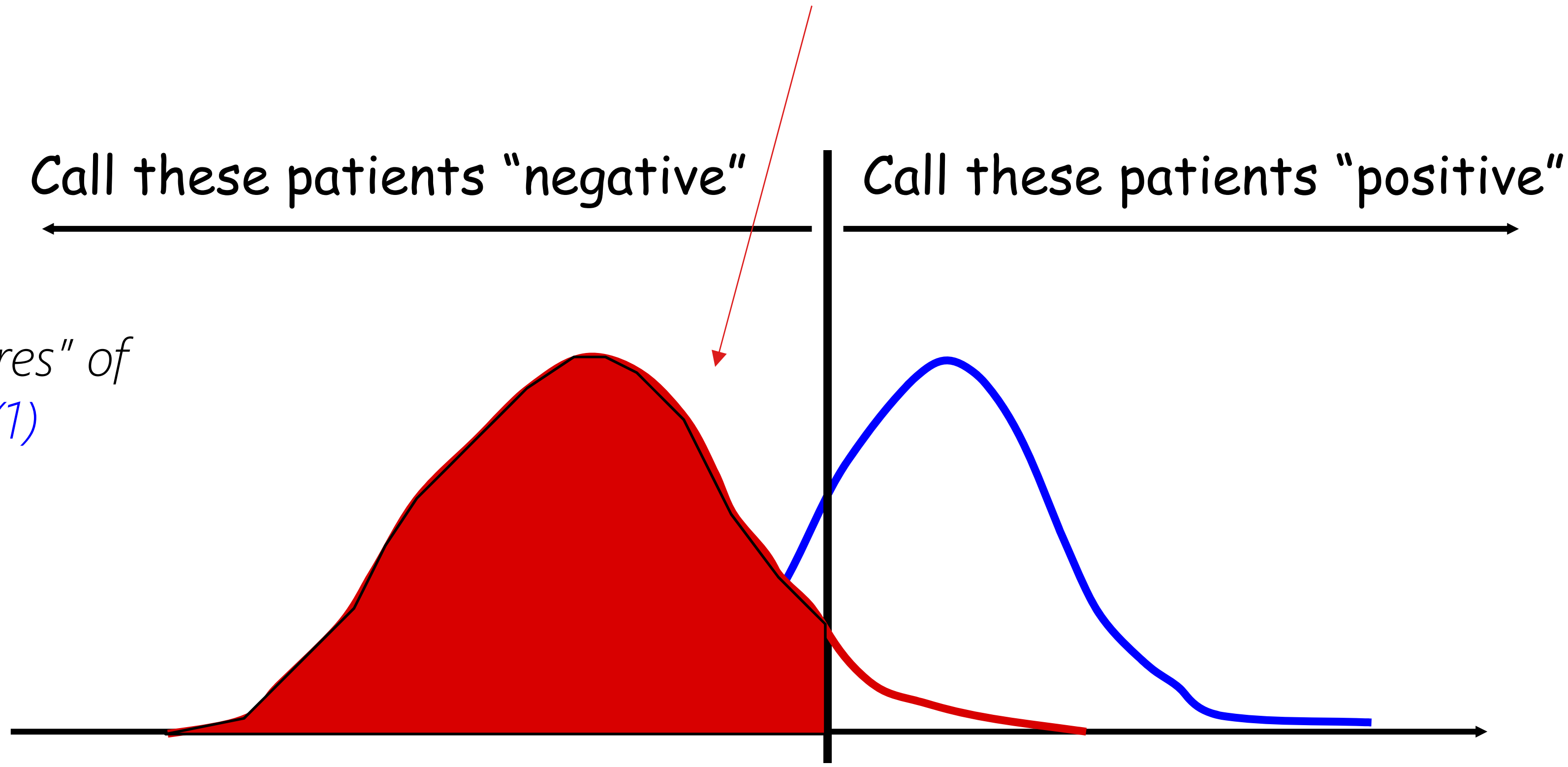
Distribution of classifier "scores" of unhealthy (-1) and healthy (1) individuals in a test set.



[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt]

Definitions: True Negatives (TN)

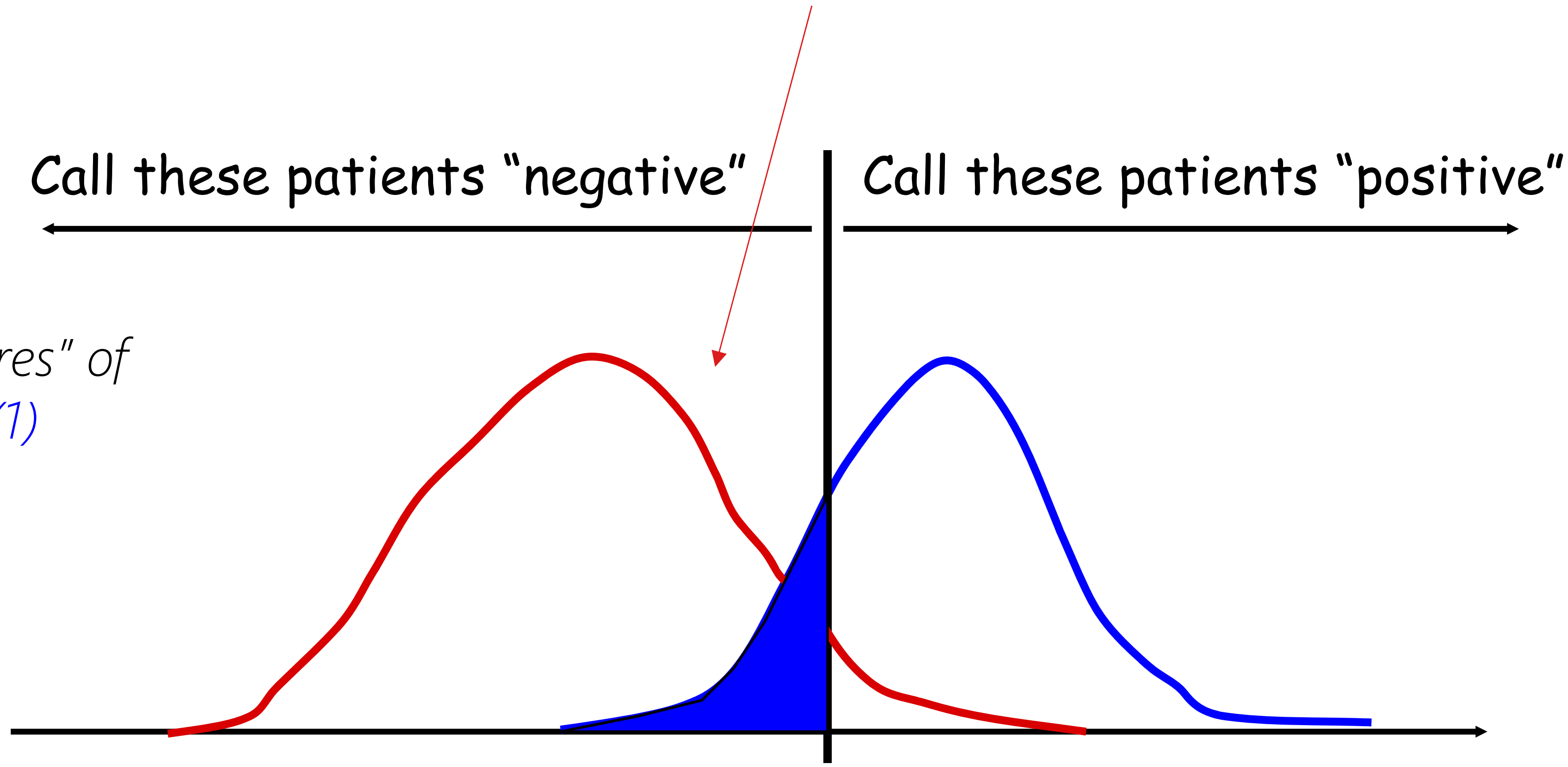
Distribution of classifier "scores" of unhealthy (-1) and healthy (1) individuals in a test set.



Classifier score/probability

[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt]

Definitions: False Negatives (FN)

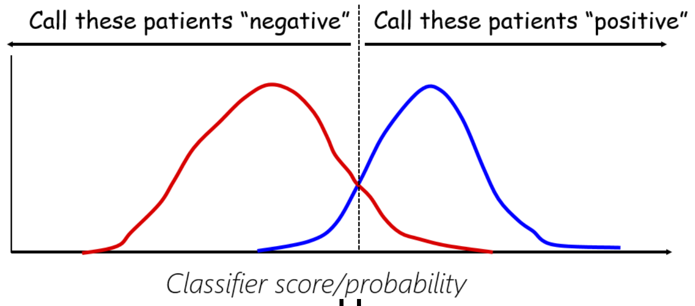


Distribution of classifier "scores" of unhealthy (-1) and healthy (1) individuals in a test set.

Classifier score/probability

[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt]

Summary of classifier decision outcomes



Once we set a *decision* threshold on the predictive score, all test data points fall into one of these four categories:

1. False Positive (FP)—person is truly a "-1" but called "1"
2. False Negative (FN)—person is truly a "1" but called "-1"
3. True Positive (TP) —person is truly a "1" and called "1"
4. True Negative (TN) —person is truly a "-1" and called "-1"

- Thus if N is total # of test points, then N=FP+FN+TP+TN
- FP and FN are mistakes when using the classifier.
- TP and TN are correct decisions when using the classifier.

Summary of classifier decision outcomes

Often we see these reported in the form of a *confusion matrix*:

		MODEL PREDICTIONS		
		Negative	*Positive*	
GROUND TRUTH	*Negative*	TN	FP	#actual negatives=TN+FP
	Positive	FN	TP	#actual positives=FN+TP

[Adapted https://hiplab.mc.vanderbilt.edu/people/malin/presentations/ROC_Curves.ppt]

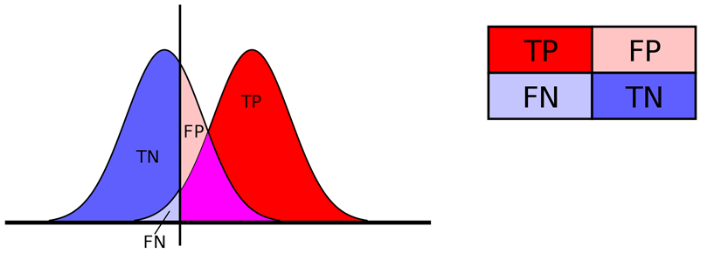
Summary of classifier decision outcomes

Rates: “normalize” by #samples who could have had that call:

- TP rate, TPR=TP/#actual positives=TP/(FN+TP), aka *Sensitivity*
- TN rate, TNR=TN/#actual negatives=TN/(TN+FP), aka *Specificity*
- FN rate, FNR=FN/#actual positives=1-TPR aka *Miss Rate*
- FP rate, FPR=FP/#actual negatives=1-TNR aka *Fall out*

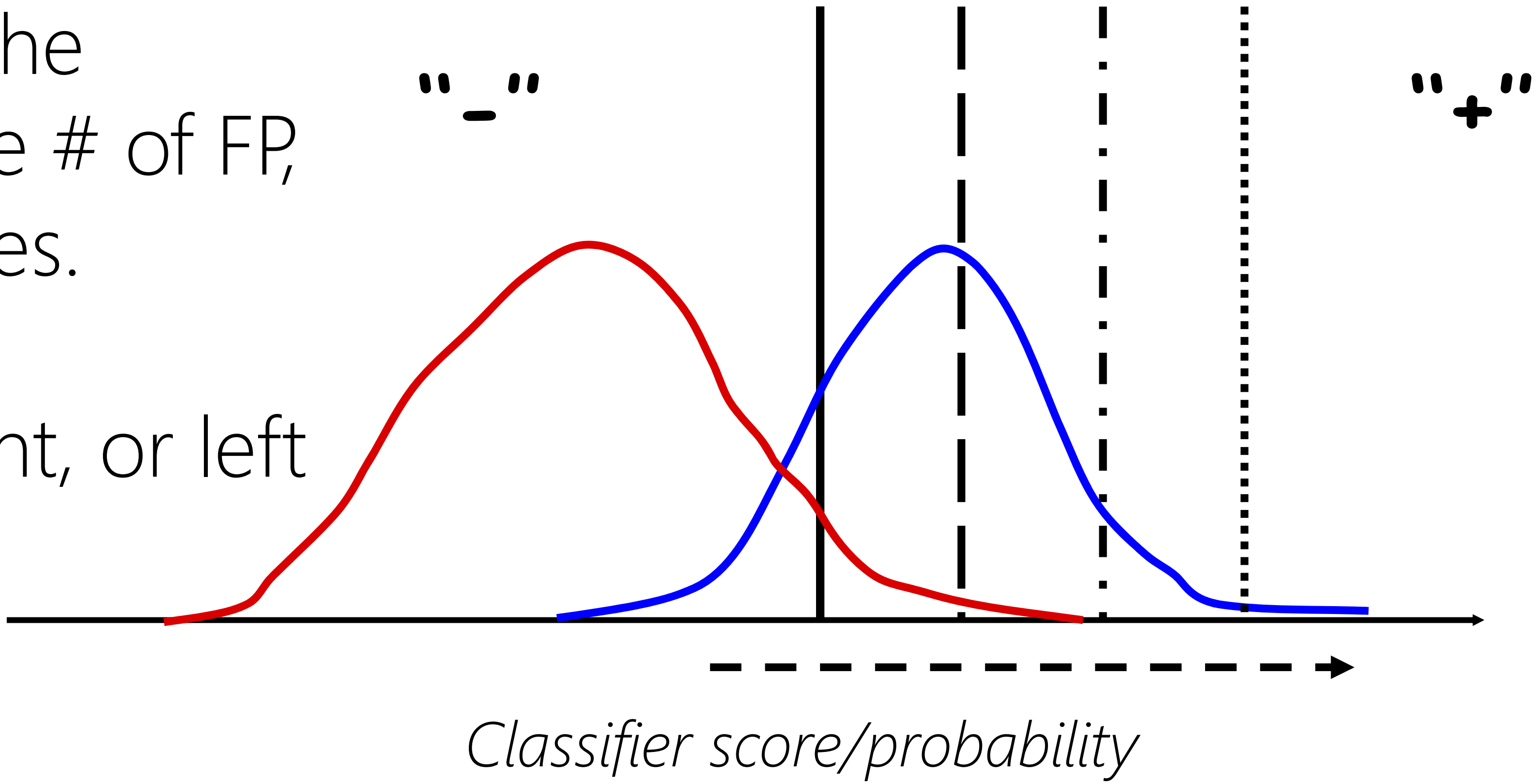
		MODEL PREDICTIONS		
		Negative	*Positive*	
GROUND TRUTH	*Negative*	TN	FP	#actual negatives=TN+FP
	Positive	FN	TP	#actual positives=FN+TP

Back to our decision threshold



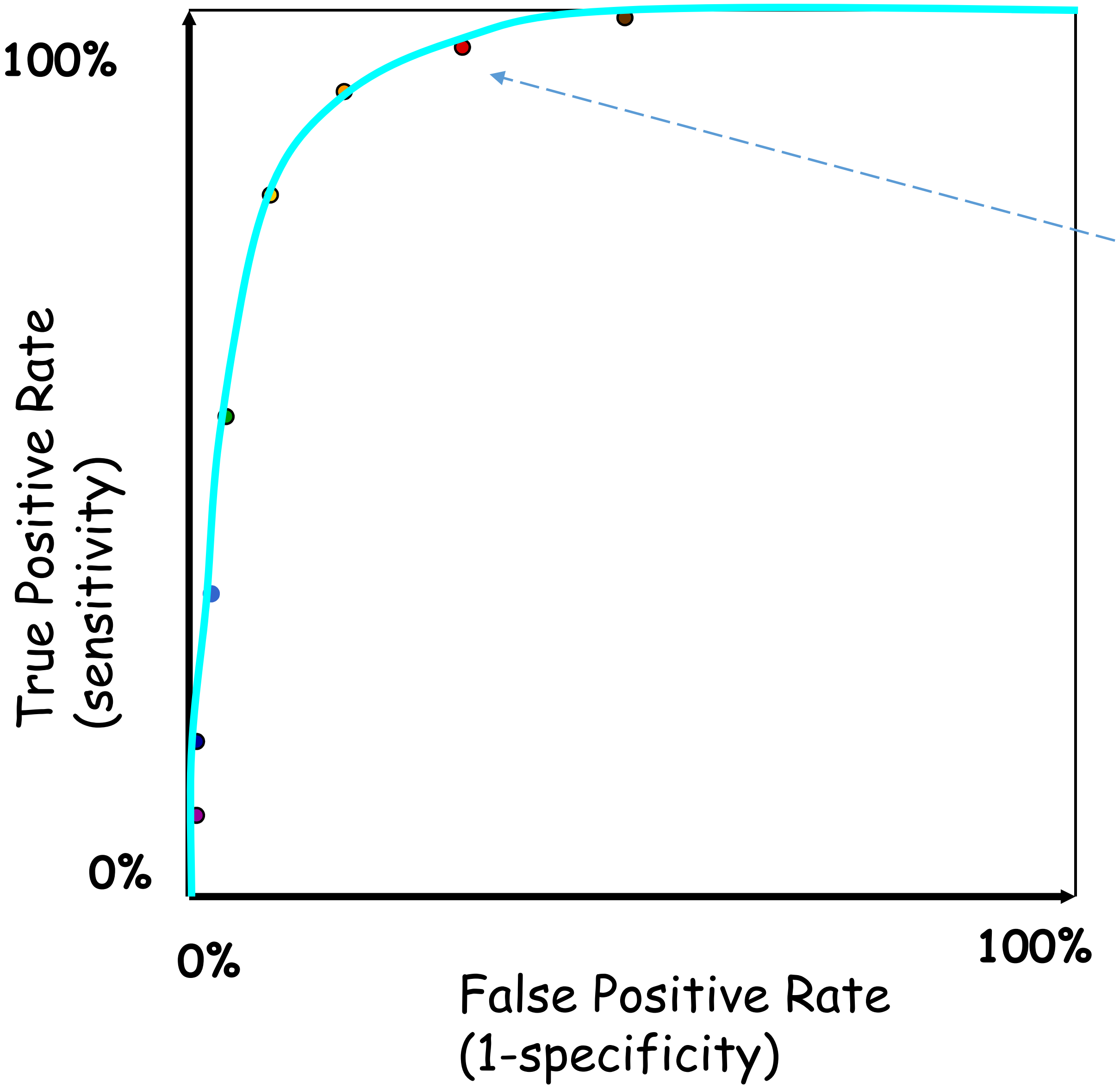
Every time we move the decision threshold, the # of FP, FN, TP and TN changes.

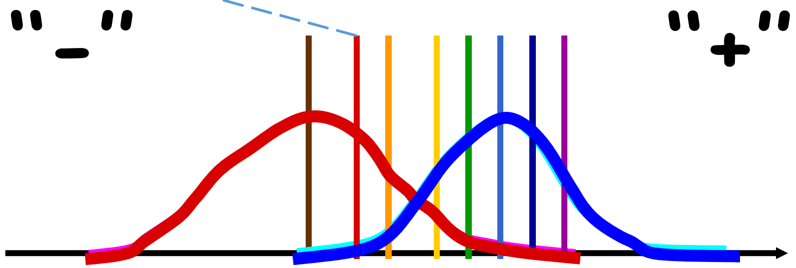
e.g. shifting to the right, or left



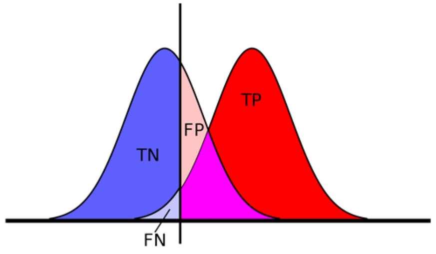
[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt], https://en.wikipedia.org/wiki/Receiver_operating_characteristic

As we shift it, we can draw out an *ROC curve*





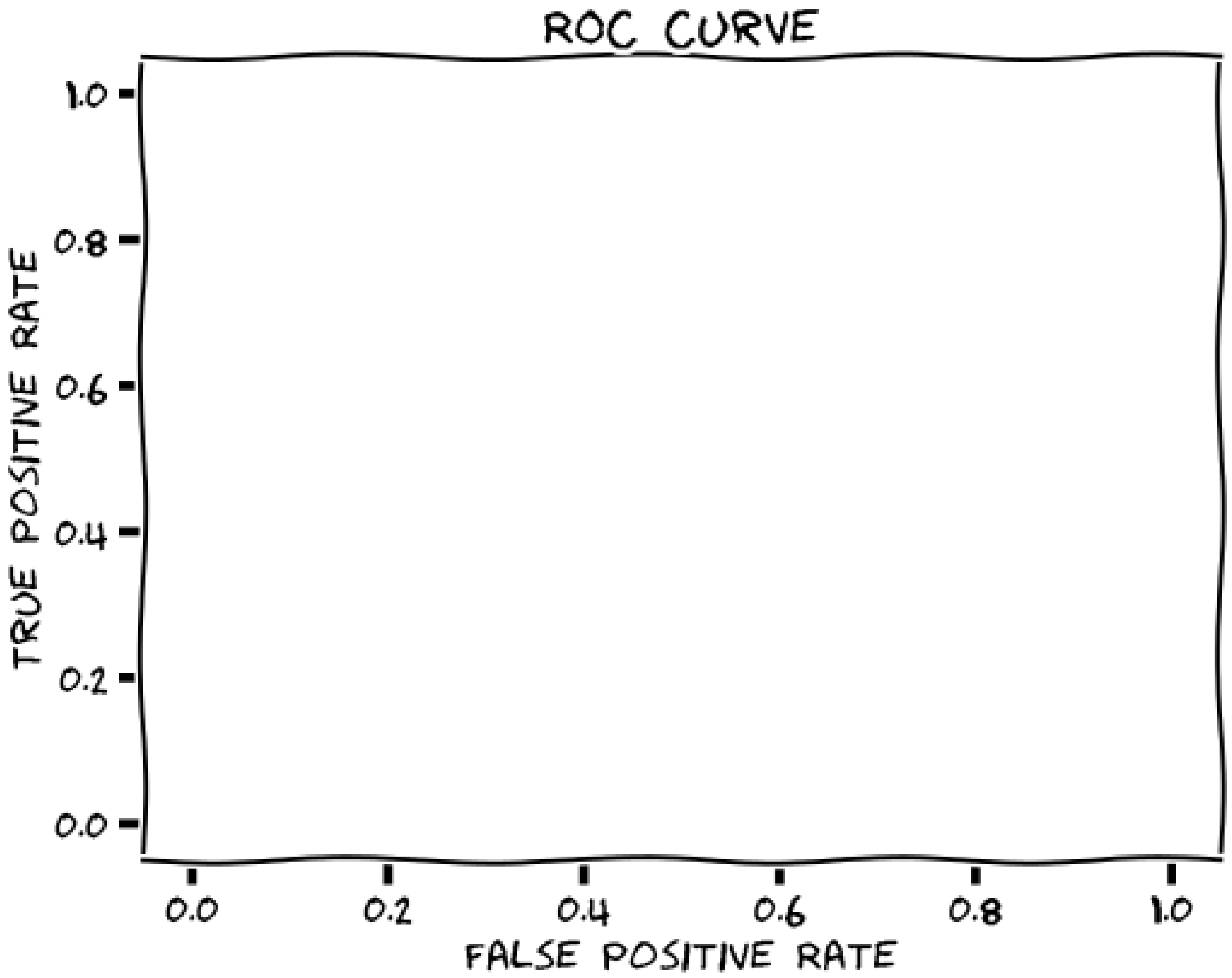
predictive distribution of classifier scores

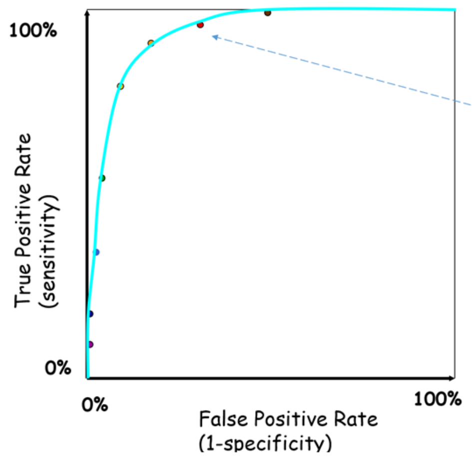


TP	FP
FN	TN

[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt], https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Comparing ROC curves across classifiers

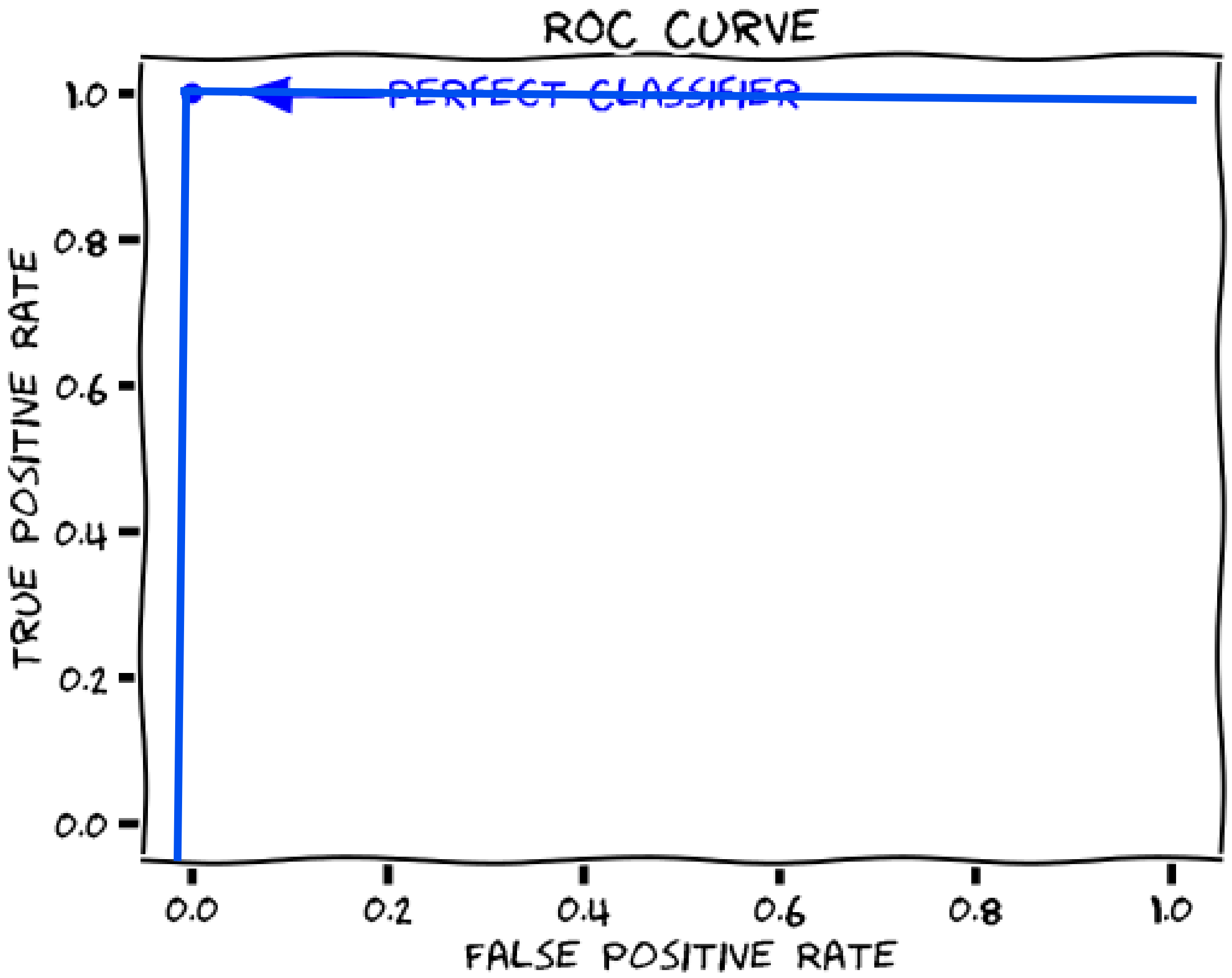


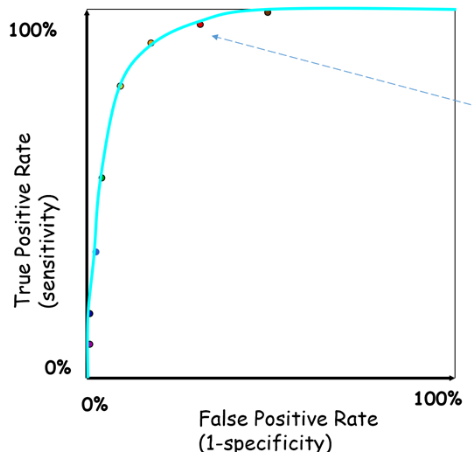


How would a perfect classifier appear on this plot?

https://en.wikipedia.org/wiki/File:Roc-draft-xkcd-style.svg

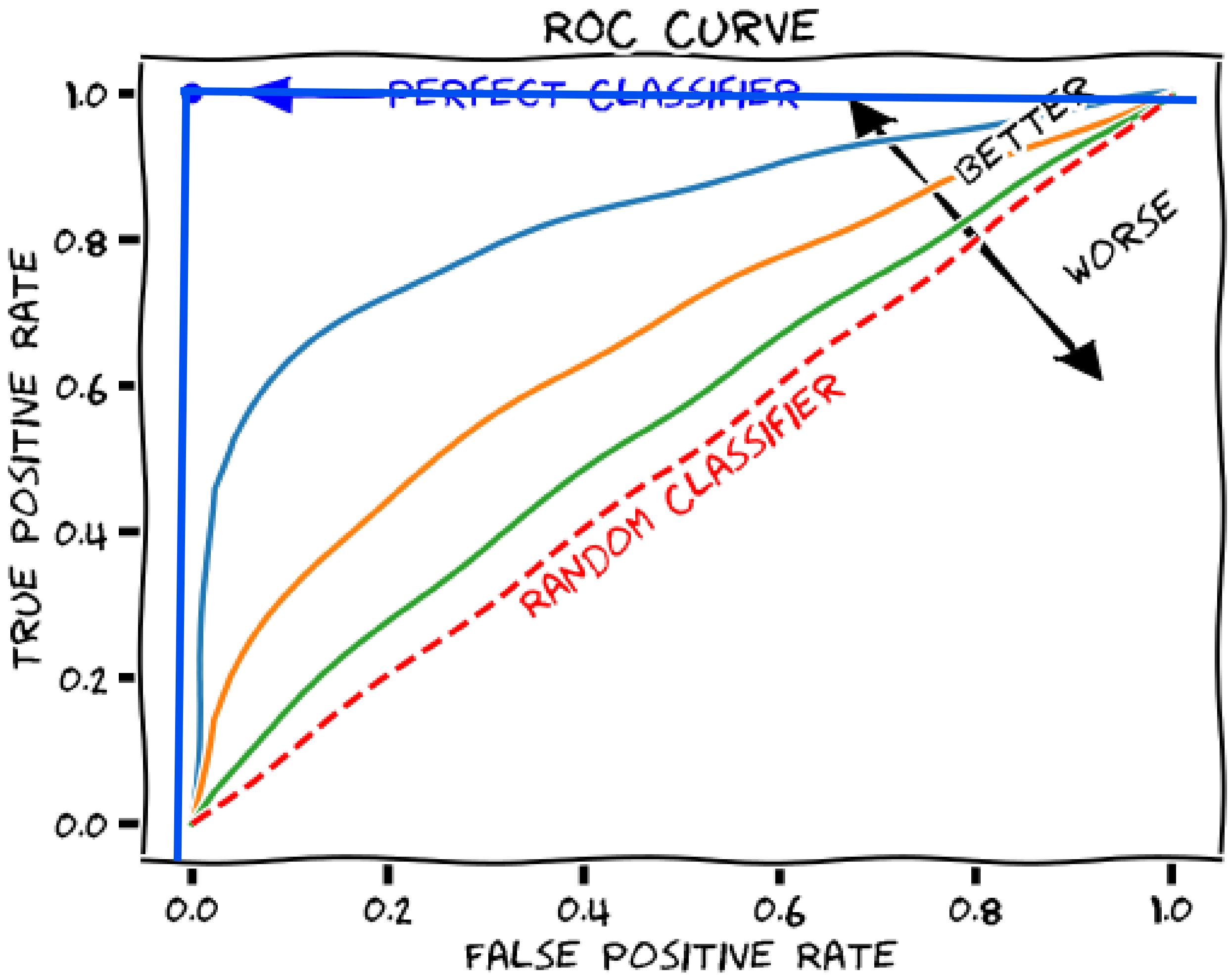
Comparing ROC curves across classifiers

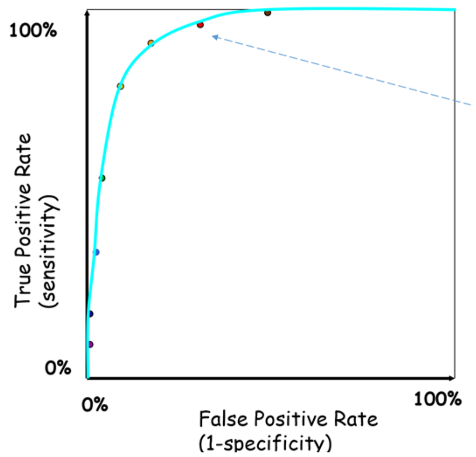




https://en.wikipedia.org/wiki/File:Roc-draft-xkcd-style.svg

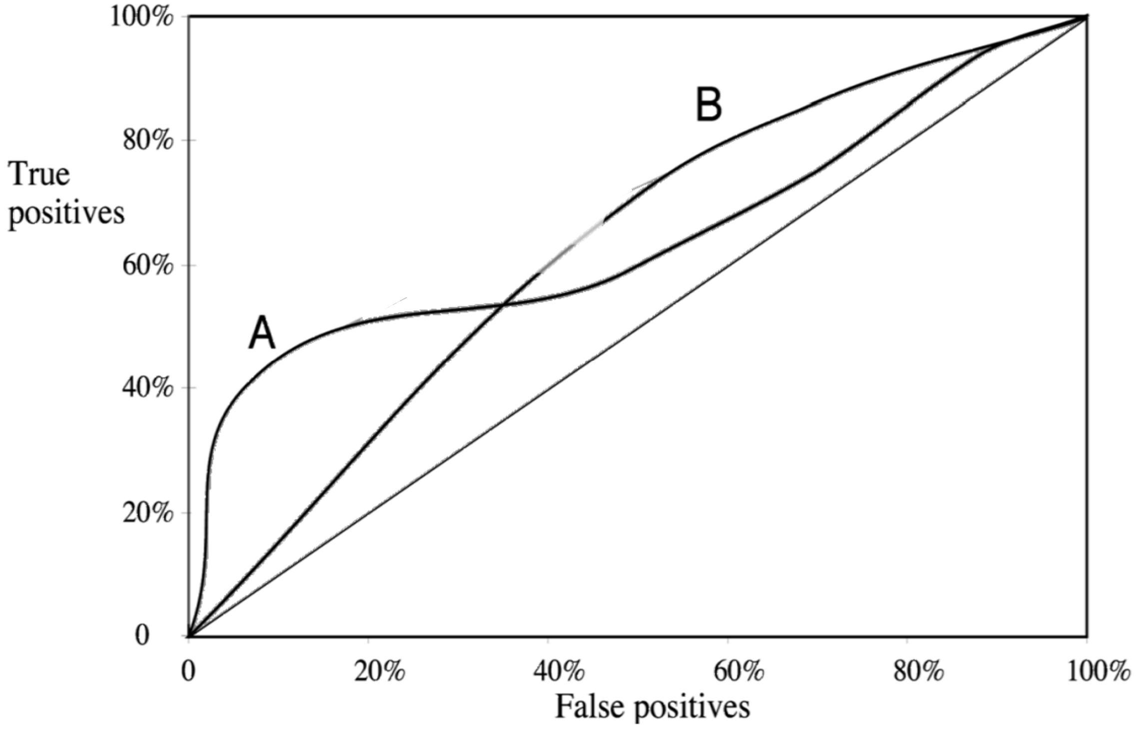
Comparing ROC curves across classifiers





https://en.wikipedia.org/wiki/File:Roc-draft-xkcd-style.svg

Which model would you choose?



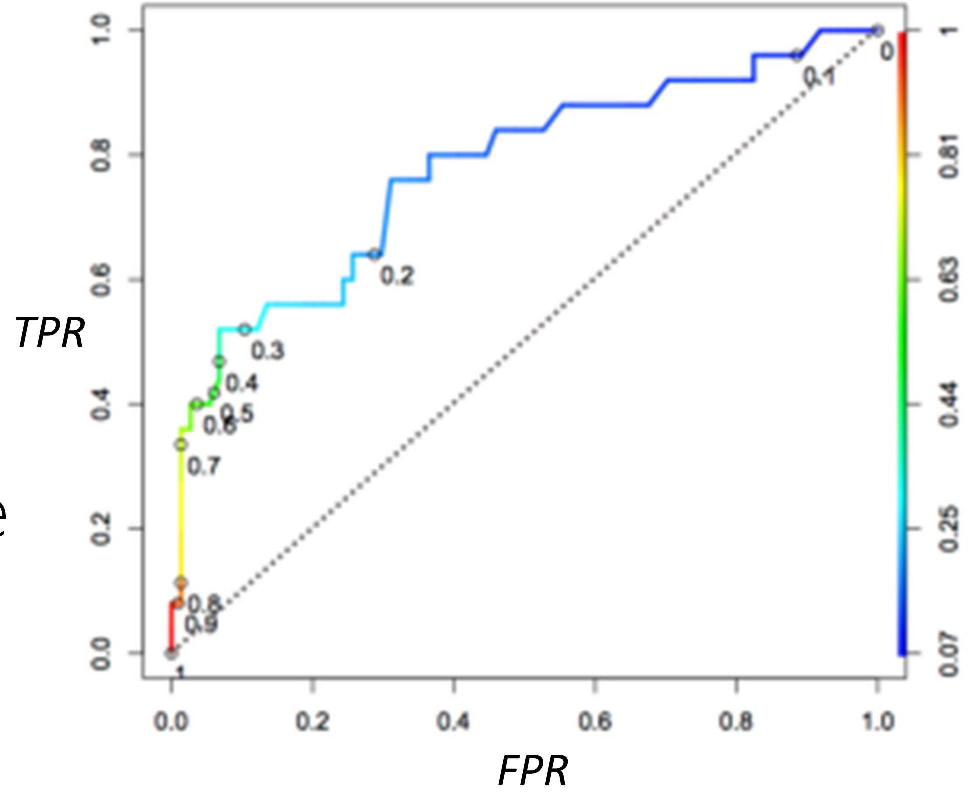
- In this example, no one classifier (A or B) *dominates* in ROC space.
- Thus we might ask what regime is most relevant to our problem.
- *e.g.* suppose you know you need few false positives, then you should pick method A, which dominates up until FPR 40%.

An algorithm for making an ROC curve

Exploit the property that any instance that is classified as + at a given threshold, will be classified as + for all lower thresholds as well:

1. Sort the test instances by decreasing score.
2. Move down the list (lowering the threshold), processing one instance at a time. For each instance,
3. Computer the TPR and FPR and add one point to the plot.

*Table does not correspond to figure

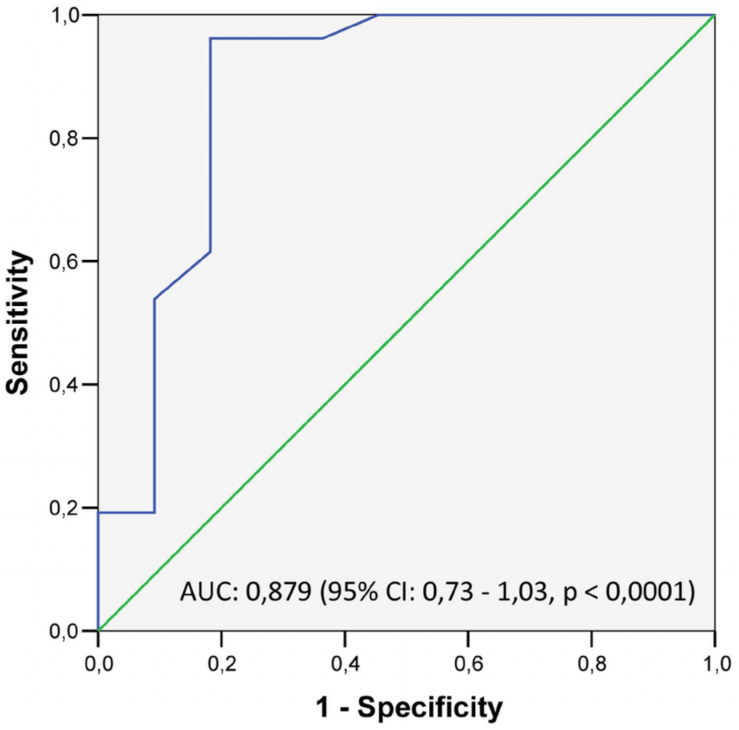


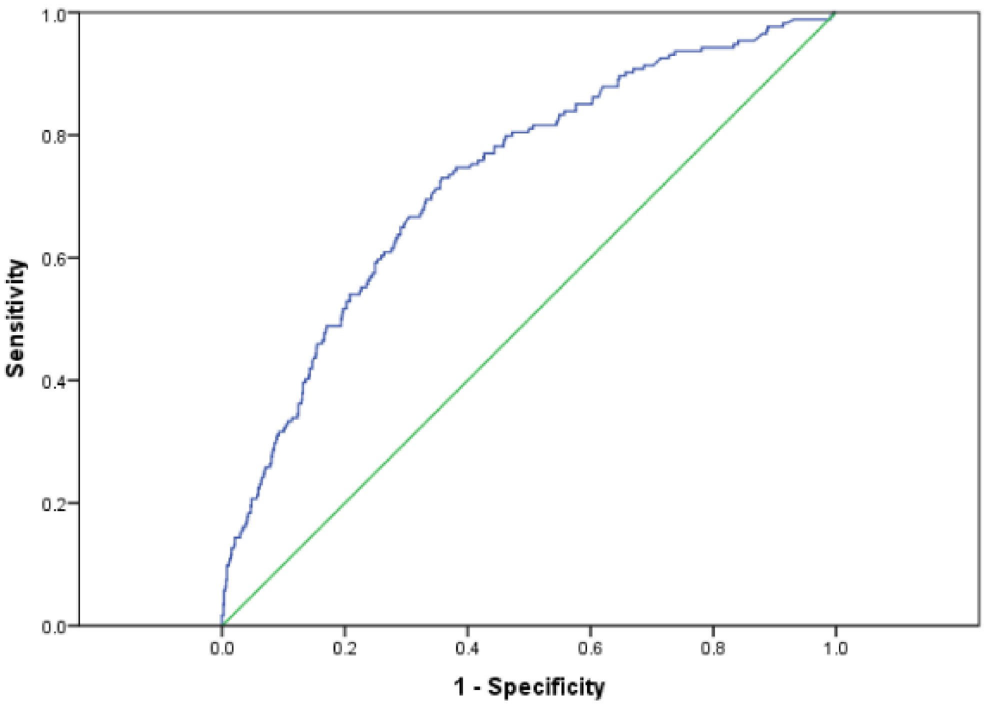
TRUTH	LOGISTIC Regression score
1	0.98
1	0.93
0	0.88
1	0.8429
0	0.81
1	0.77
0	0.7527
0	0.7194
1	0.59
0	0.42
0	0.0839

https://lukeoakdenrayner.wordpress.com/2018/01/07/the-philosophical-argument-for-using-roc-curves/

An algorithm for making an ROC curve

- Smoothness of the ROC curve is dependent on the # of points in it
- Restricted by # of test points and uniqueness of scores:

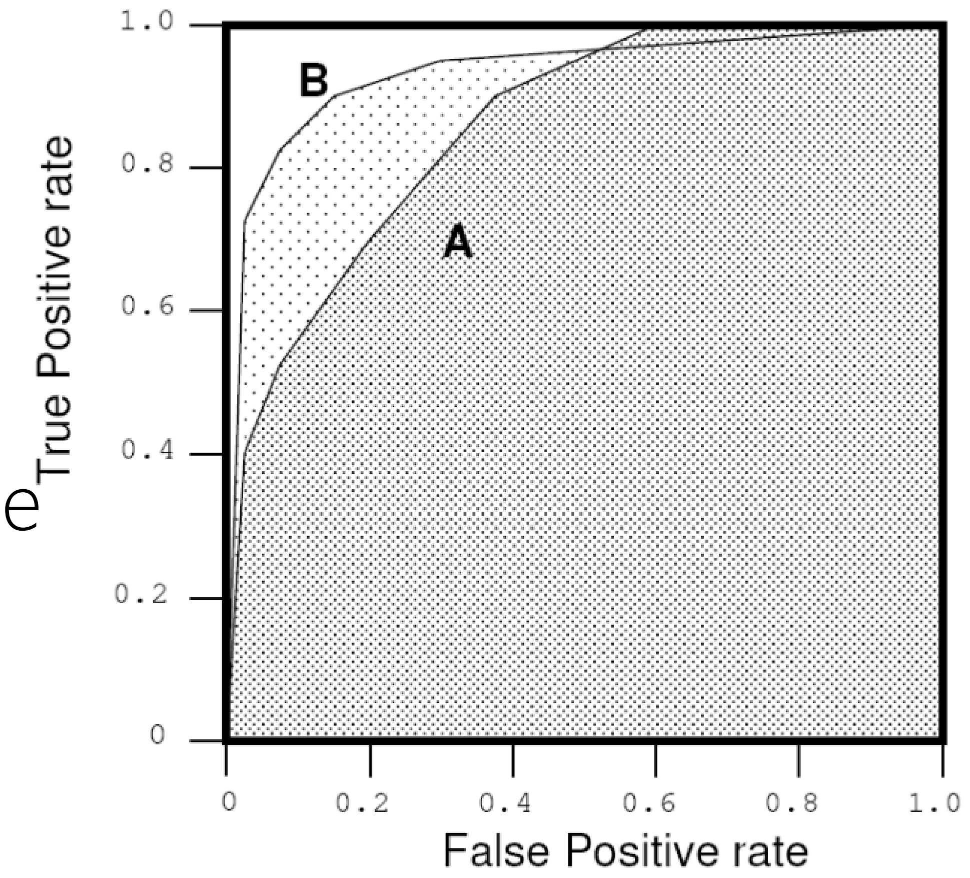


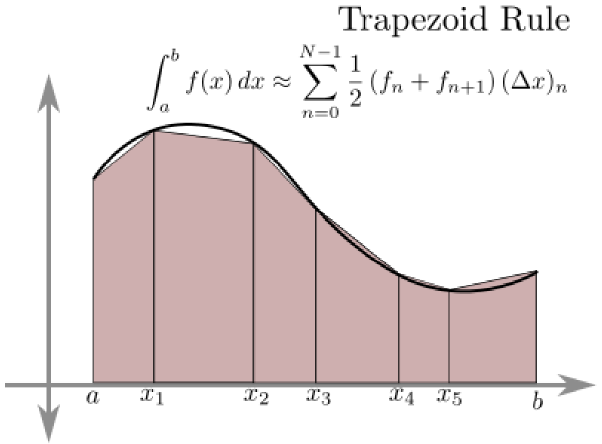


https://www.theanalysisfactor.com/what-is-an-roc-curve, https://www.researchgate.net/figure/The-Receiver-Operating-Characteristic-ROC-curve-of-different-OUES-cut-off-points_fig2_263971330

Summarizing ROCs with the Area Under the Curve (AUC)

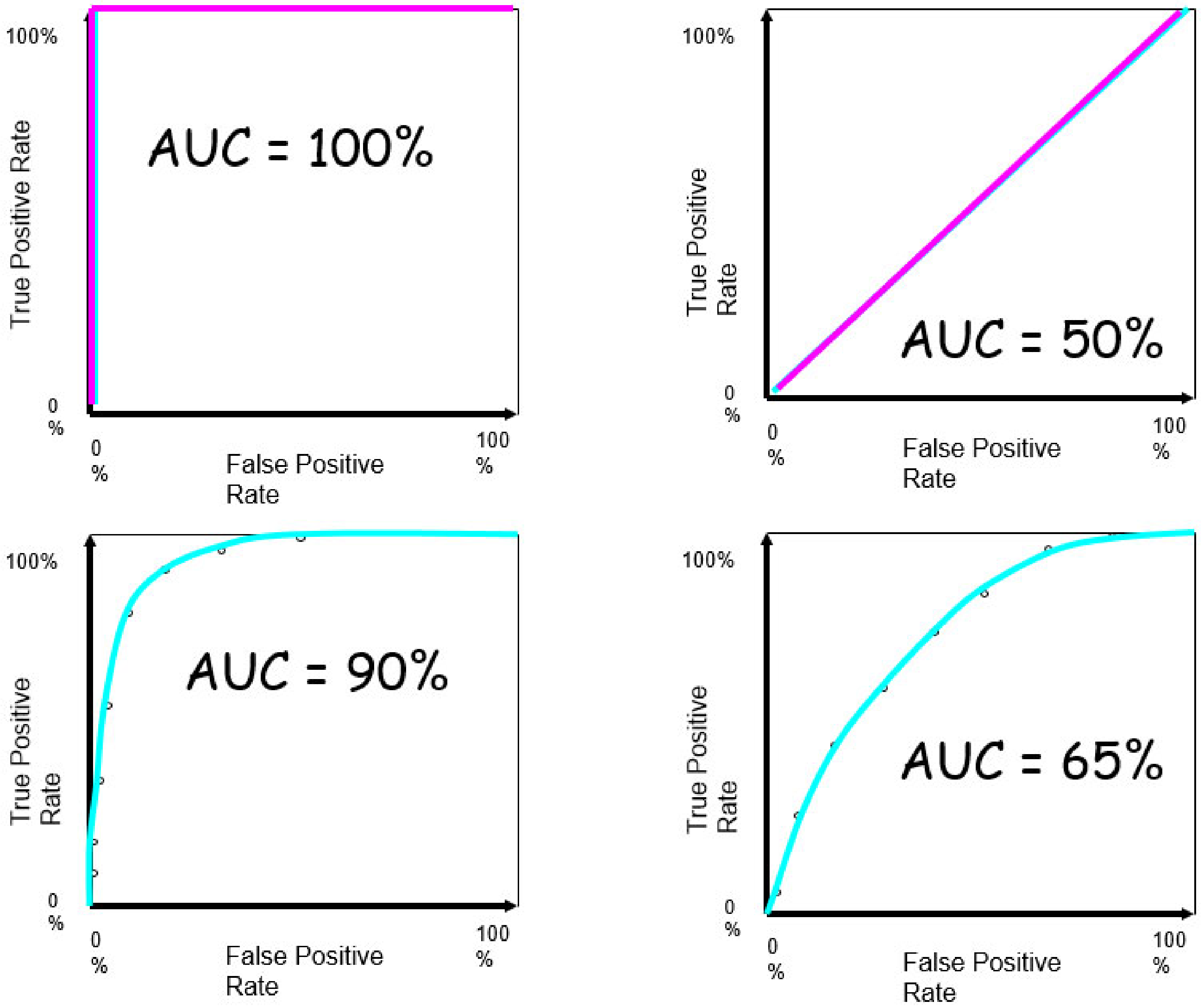
- AUC: often used to compare classifiers.
- The bigger the AUC the better.
- AUC can be computed by a slight modification to the algorithm for constructing ROC curves—basically a simple form of integration to compute the area under the curve.





https://web.uvic.ca/~maryam/DMSpring94/Slides/9_roc.pdf

Summarizing ROCs with the Area Under the Curve (AUC)

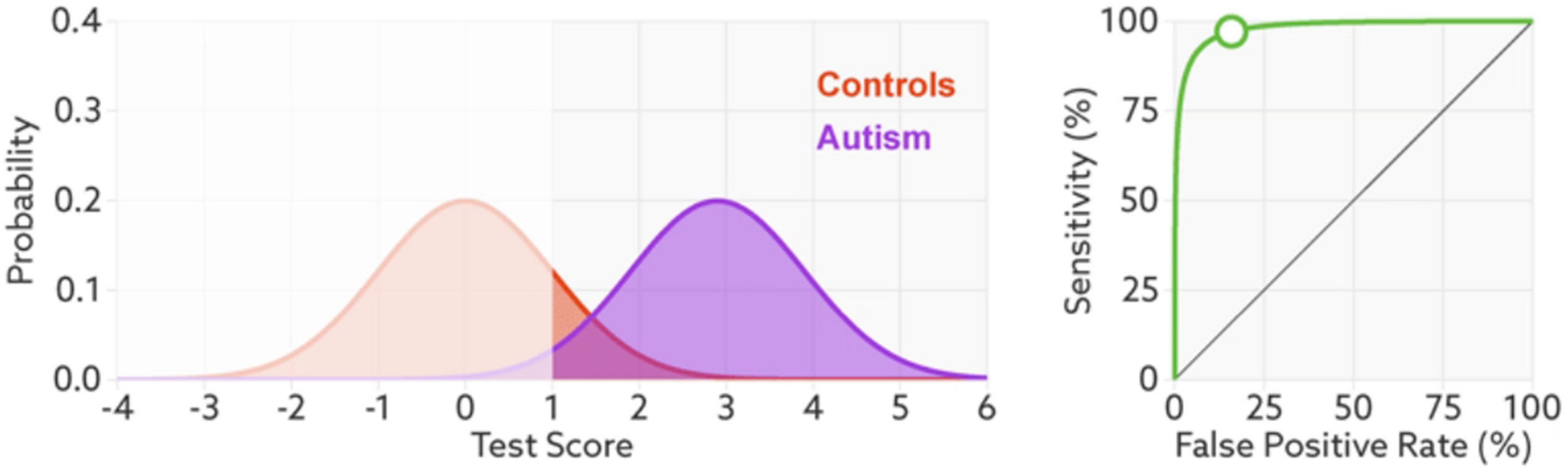


The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive sample higher than a randomly chosen negative sample.

[Adapted from http://www.lausanne.isb-sib.ch/~darlene/ms/SIB-ROC.ppt]

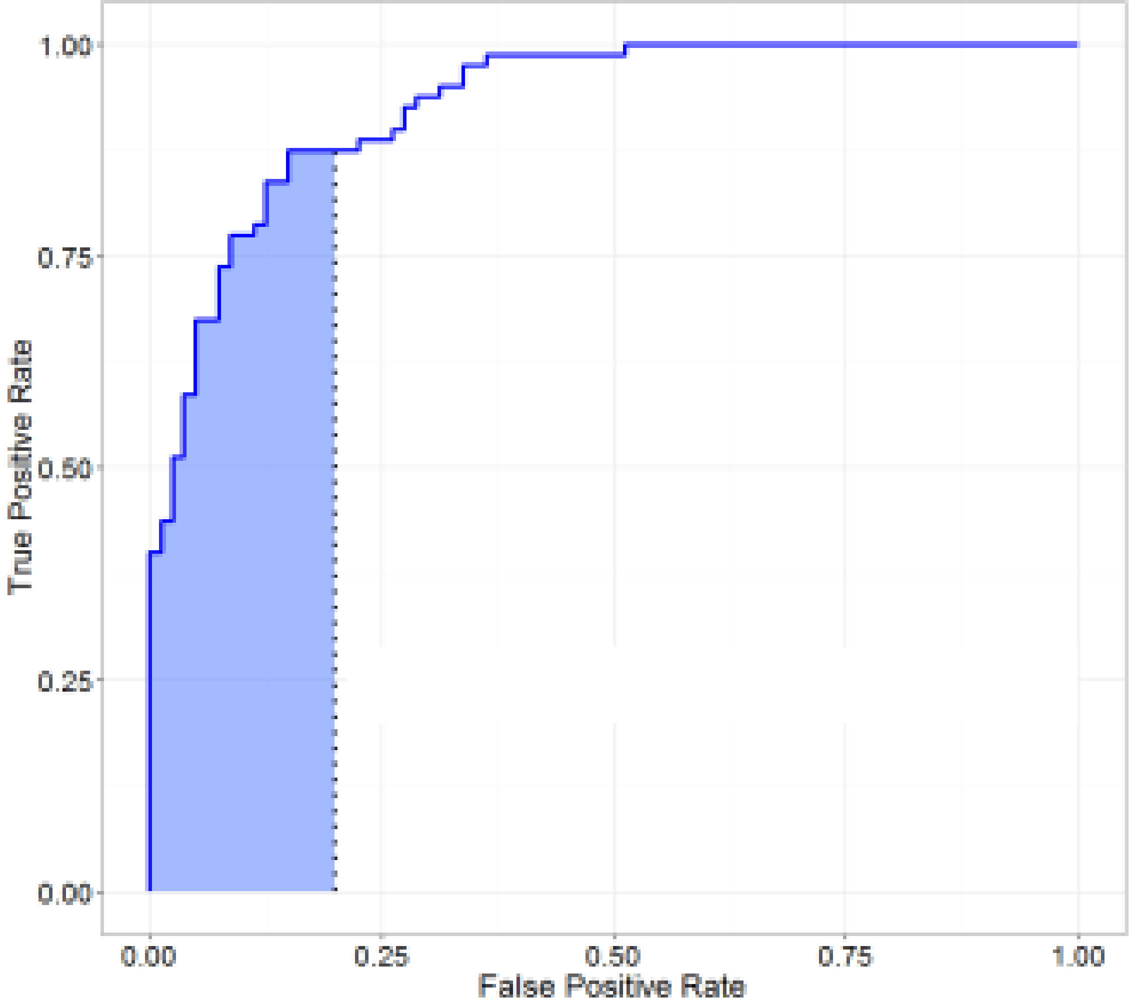
Visualization of score distributions wrt ROCs & AUC

1. Sweeping a threshold through the predictions for one classifier traces out the ROC curve.
2. The more separated the distribution of scores between the two classes, the larger the AUC.



https://www.spectrumnews.org/opinion/viewpoint/quest-autism-biomarkers-faces-steep-statistical-challenges/

Partial Area Under the Curve (AUC)



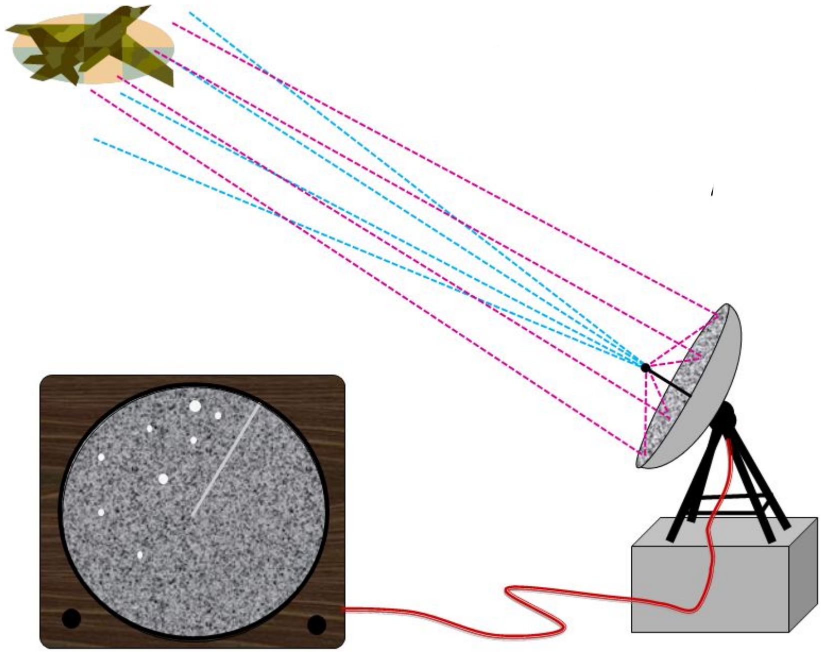
Sometimes we know we would never operate with a FPR above some amount, and so we compute a *partial* AUC, here *AUC(0.2)*.

- *e.g.* deciding on medical intervention like chemotherapy
- *e.g.* deciding to spend lots of $$$ in follow up biology experiments.

https://www.epeter-stats.de/partial-auc-support-added-fbroc-0-4-0/

History of ROC curves

- "The ROC curve was first developed by electrical engineers and radar engineers during World War II (1939-45) for detecting enemy objects in battle fields and was soon introduced to psychology to account for perceptual detection of stimuli."
- "ROC analysis since then has been used in medicine, radiology, biometrics, and other areas for many decades and often used in machine learning and data mining research and applications."



from http://en.wikipedia.org/wiki/Receiver_operating_characteristic

More on ROC curves

ROC curves are insensitive to the balance of classes in the test set (because FPR and FNR are insensitive quantities).

- TP rate, TPR=TP/#actual positives=TP/(FN+TP)
- TN rate, TNR=TN/#actual negatives=TN/(TN+FP)

- To obtain classification accuracy from an ROC, need to know the balance (i.e. ratio of # actual positives to # actual negatives in the test set).
- Knowing this we can find a point on the graph with optimal classification accuracy.
- Sometimes we use Precision-Recall curves instead of ROC because desire sensitivity to the balance of classes.

Summary of classifier decision outcomes

Rates: “normalize” by #samples who could have had that call:

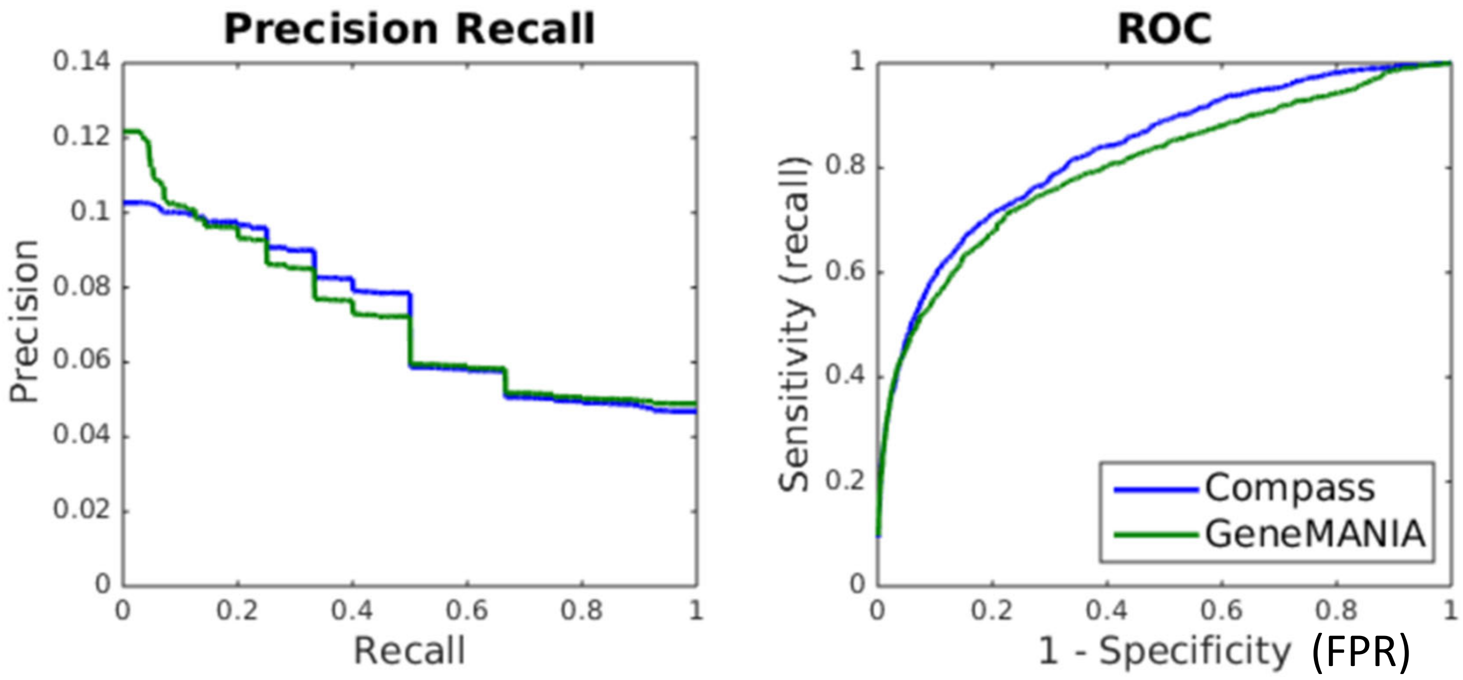
- TP rate, TPR=TP/#actual positives=TP/(FN+TP), aka “Sensitivity”
- TN rate, TNR=TN/#actual negatives=TN/(TN+FP), aka “Specificity”
- FN rate, FNR=FN/#actual positives=1-TPR aka “Miss Rate”
- FP rate, FPR=FP/#actual negatives=1-TNR aka “Fall out”
- Precision=TP/(#predicted positive)=TP/(TP+FP)—this now depends on class balance in test set.
- Recall=TPR=Sensitivity

		MODEL PREDICTIONS		
		Negative	*Positive*	
GROUND TRUTH	*Negative*	TN	FP	#actual negatives=TN+FP
	Positive	FN	TP	#actual positives=FN+TP

Precision Recall Curves

- Wrt ROC: replace FPR with Precision and flip the axes.

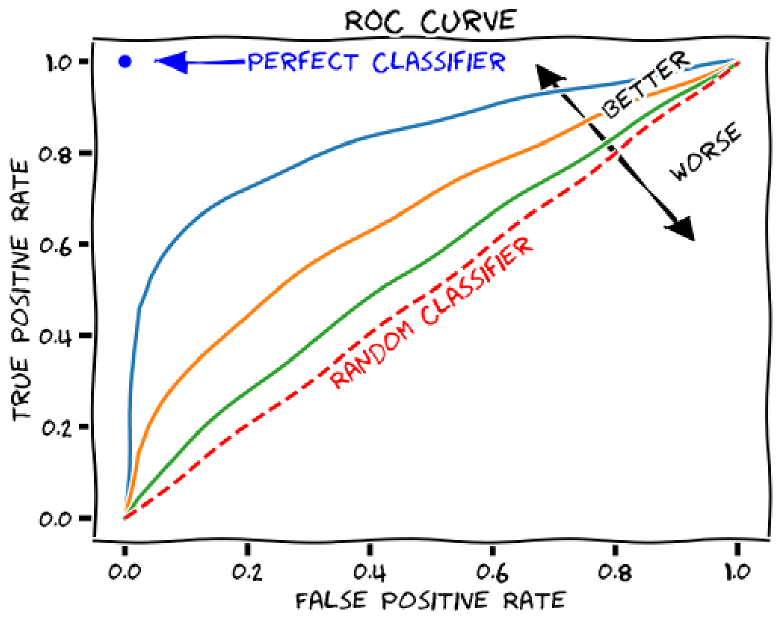
- ROC curves useful when we want invariance to class distribution.
- Precision-recall curves useful when care about (and know) the balance of the classes at test time.



https://www.researchgate.net/figure/Average-precision-recall-and-ROC-curves-for-the-GO-rollback-benchmark-The-curves_fig2_281172305

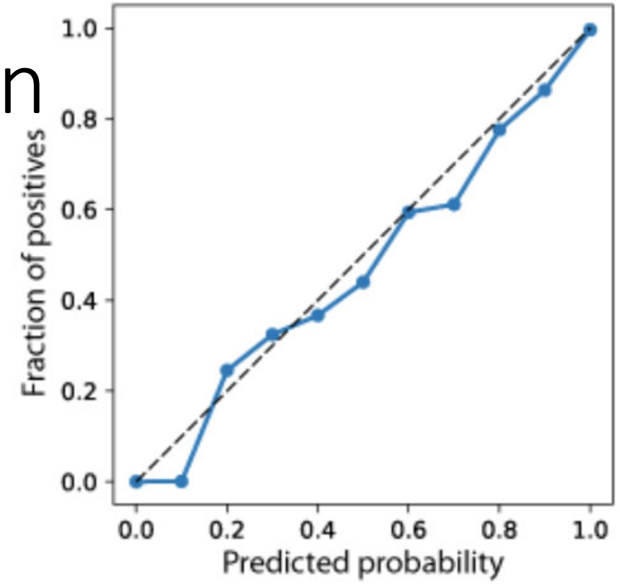
Summary of ROC curves and their utility

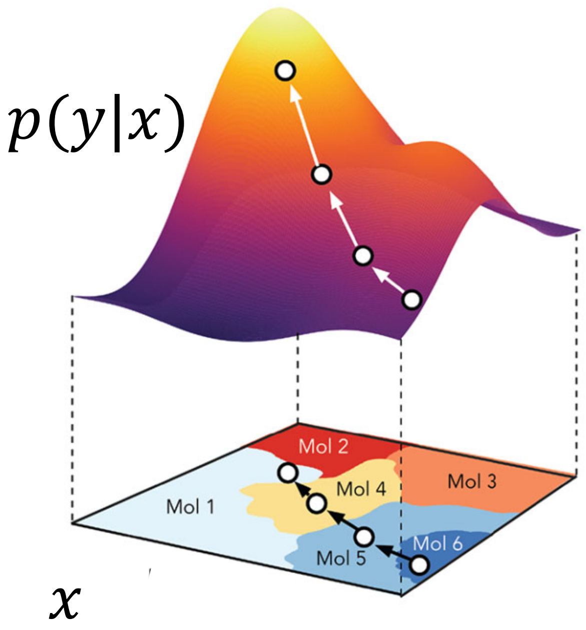
- Gives more nuanced understanding than counting the # of misclassifications.
- Does not require a decision threshold.
- Summarizes performance of binary classification models, across all possible trade-offs in decision making FPR and FNR.
- Does not care about model calibration (can be a pro or a con).
- Can compare classifiers by comparing their AUC summary, or using the entire ROC curves, or just part of curve for AUC/ROC.



If you care about (probabilistic) model calibration

- Then you should NOT use ROC curves to evaluate, because they don't care about "calibrated uncertainty" of the predictions.
- Calibrated uncertainty can be very important in medical applications, such as to determine treatments, or administering invasive diagnostics.
- Calibration also come into ML-based design---e.g. in small molecule engineering (e.g. use a predictive model to design the best binder to drug target).
- Also in "active learning".





ACS Cent. Sci. 2018, 4, 268–276

Many other evaluations—dictated by the domain of application

- Predict a Ranking (of webpages)
  - Users only look at top 4
  - Sort by f(x|w,b)

- Precision @4 =1/2
  - Fraction of top 4 relevant

- Recall @4 =2/3
  - Fraction of relevant in top 4

- Top of Ranking Only!

Image Source: http://pmtk3.googlecode.com/svn-history/r785/trunk/docs/demos/Decision_theory/PRhand.html

If we care about these metrics, why not use them as a loss function to train the model?

- These losses are often not differentiable everywhere (e.g. AUC and others have hard thresholding).
- Can't use mini-batch (ranking requires entire test set), hence can't use SGD.
- Some niche attempts to make metrics differentiable, but in practice, rarely used.

<u>Learning</u> cost-sensitive classifiers

- In regular classification (all we have seen so far), we treat all types of misclassification errors the same (e.g. in using a likelihood loss/cost function).
- A more general setting is to learn *cost-sensitive* classifiers where we allow for different types of errors to have more or less impact.

Patient

Model	Loss Function	Has Cancer	Doesn't Have Cancer
	Predicts Cancer	Low	Medium
	Predicts No Cancer	**OMG Panic!**	Low

[table from Yisong Yue]

<u>Learning</u> cost-sensitive classifiers

- In regular classification (all we have seen so far), we treat all types of misclassification errors the same (e.g. in using a likelihood loss/cost function).
- A more general setting is to learn *cost-sensitive* classifiers where we allow for different types of errors to have more or less impact.

Patient

Model	Loss Function	Has Cancer	Doesn't Have Cancer
	Predicts Cancer	Low	Medium
	Predicts No Cancer	**OMG Panic!**	Low

[table from Yisong Yue]

Optimizing for Cost-Sensitive Loss

- No universally accepted way, but a common and simple one is to use a "Cost Balancing" loss (effectively you are "rebalancing" the training data by your costs):

$$\underset{w,b}{\operatorname{argmin}}\left(1000\sum_{i:y_i=1} L\big(y_i, f(x_i \mid w,b)\big) + \sum_{i:y_i=-1} L\big(y_i, f(x_i \mid w,b)\big)\right)$$

user-defined cost matrix

Loss Function	Has Cancer	Doesn't Have Cancer
Predicts Cancer	0	1
Predicts No Cancer	1000	0

[table from Yisong Yue]

Extra