Performance measures

Reminders/comments

- Reading week next week
- Today: more about designing experiments
  - for your mini-project
- Each graduate student will be randomly assigned two mini-projects
- As a reminder, you can work in pairs

Reminder: Experimental set-up

- Performance measures
- Sampling: How to obtain multiple samples of performance?
- Making conclusions: Statistical significance tests
- Careful statistical work done on executing empirical studies; pros and cons to each
  - for a nice reference, see Evaluating Learning Algorithms: A Classification Perspective (http://www.mohakshah.com/tutorials/icml2012/Tutorial-ICML2012/Tutorial_at_ICML_2012.html); slides in this lecture use some of the material there
  - "Prediction error estimation: a comparison of resampling methods"

Statistical significant test

- Can the observed results be attributed to real characteristics of the learner under scrutiny or are they observed by chance?

- Hypothesis testing:
    - State a null hypothesis, e.g., the expected errors of two classifiers is equivalent
    - Choose a statistical significance test to reject the null hypothesis; failing to reject the null hypothesis does not mean we accept it
    - Rejecting the null hypothesis gives us some confidence in the belief that our observations did not occur merely by chance.

Types of errors

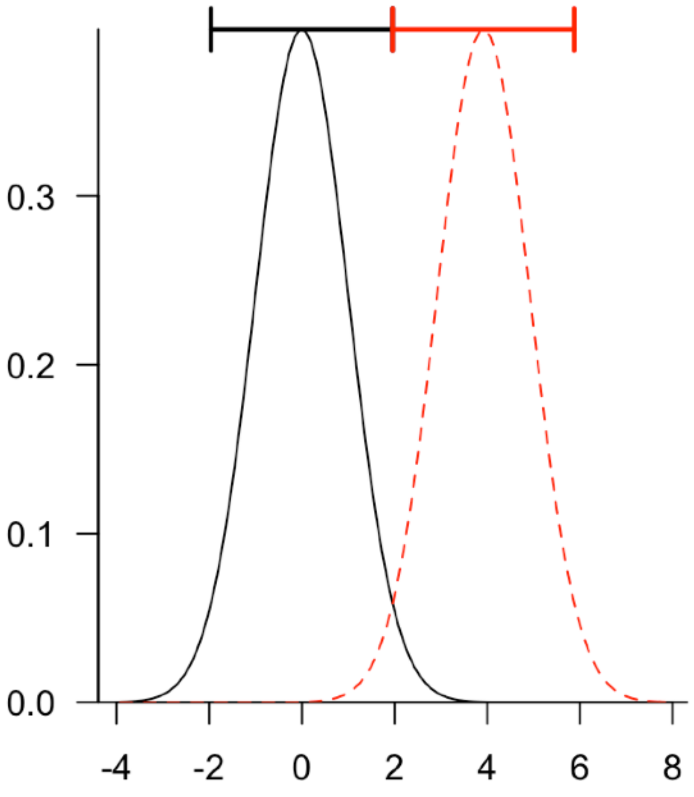
- Type 1 error: rejecting the null hypothesis when it is true
  - could occur if you select alpha too large (e.g., alpha = 0.05)
  - could occur if you violate assumptions, e.g., equal variances
- Type 2 error: failure to reject the null hypothesis when it is false
  - these usually occur if we select a test with insufficient power, e.g., just checking if intervals overlap

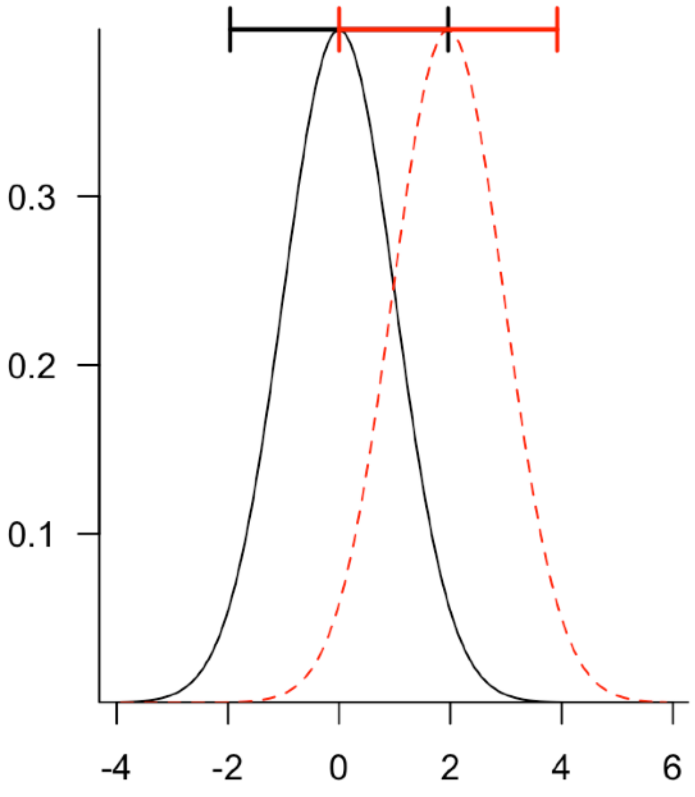
Comparing two algorithms on a single domain

- Imagine you have N independent test-samples, giving N paired measures of error for the two algorithms

- Simplest (not very powerful) strategy:
  - compute two (95%) confidence intervals for the means
  - if the two intervals do not overlap, means are significantly different

different



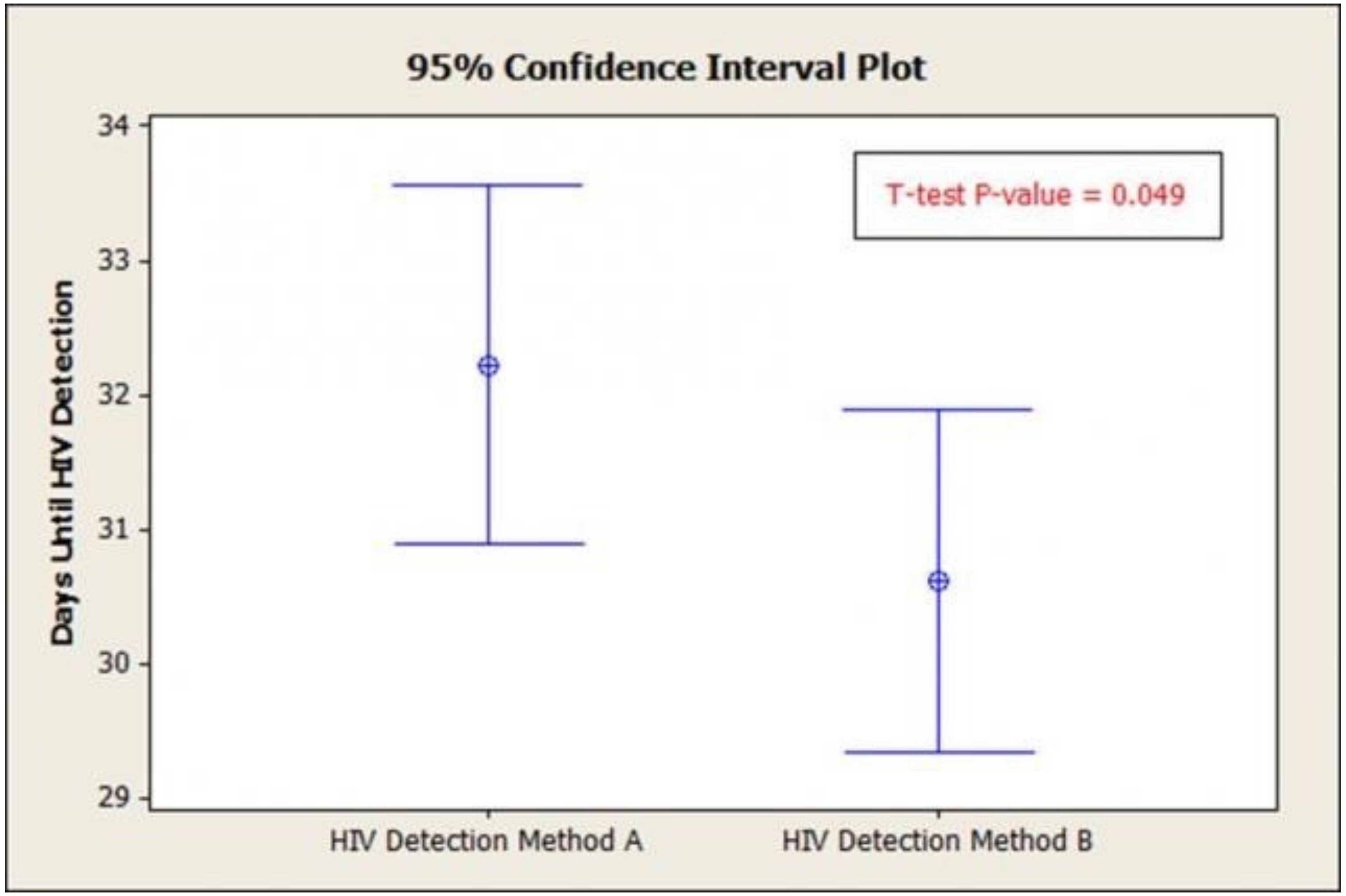


unknown

More powerful strategy: t-test

- Confidence intervals may overlap, but the means may still be statistically different

- Paired t-test enables a more powerful comparison
  - more ability to reject the null hypothesis



* image from http://blog.minitab.com/blog/real-world-quality-improvement/common-statistical-mistakes-you-should-avoid

Sampling

- How do we get independent samples of test error? And why?
    - want to get a measure of generalization performance for an algorithm
    - want to compare algorithms
    - want to do hyperparameter selection (e.g., regularization parameters)
- We can only get approximate samples of true expected error
- Subsampling approaches
    - Cross-validation (CV)
    - Repeated subsampling: Monte carlo CV
    - Bootstrap resampling

Your goal

- Try to keep the biases in mind when designing your experiment
- You will not be able to obtain a perfect experiment design
- But, you can be careful about
  - introducing really obviously fixable biases
  - picking inappropriate algorithms
  - giving some algorithms an unfair advantage
  - picking inappropriate error measures

Exercise: comparing two networks

- Imagine you are comparing an NN with a single hidden layer, one with relu on the first layer and one with sigmoid
    - lets say the outputs are linear
- For your dataset, you've noticed that previously 30 hidden units worked well with relu, and 20 for sigmoid
- You run the two (with a fair resampling strategy), and find relu does better
- Can you conclude that relu is a better activation choice for this dataset?

Recall our sampling approaches

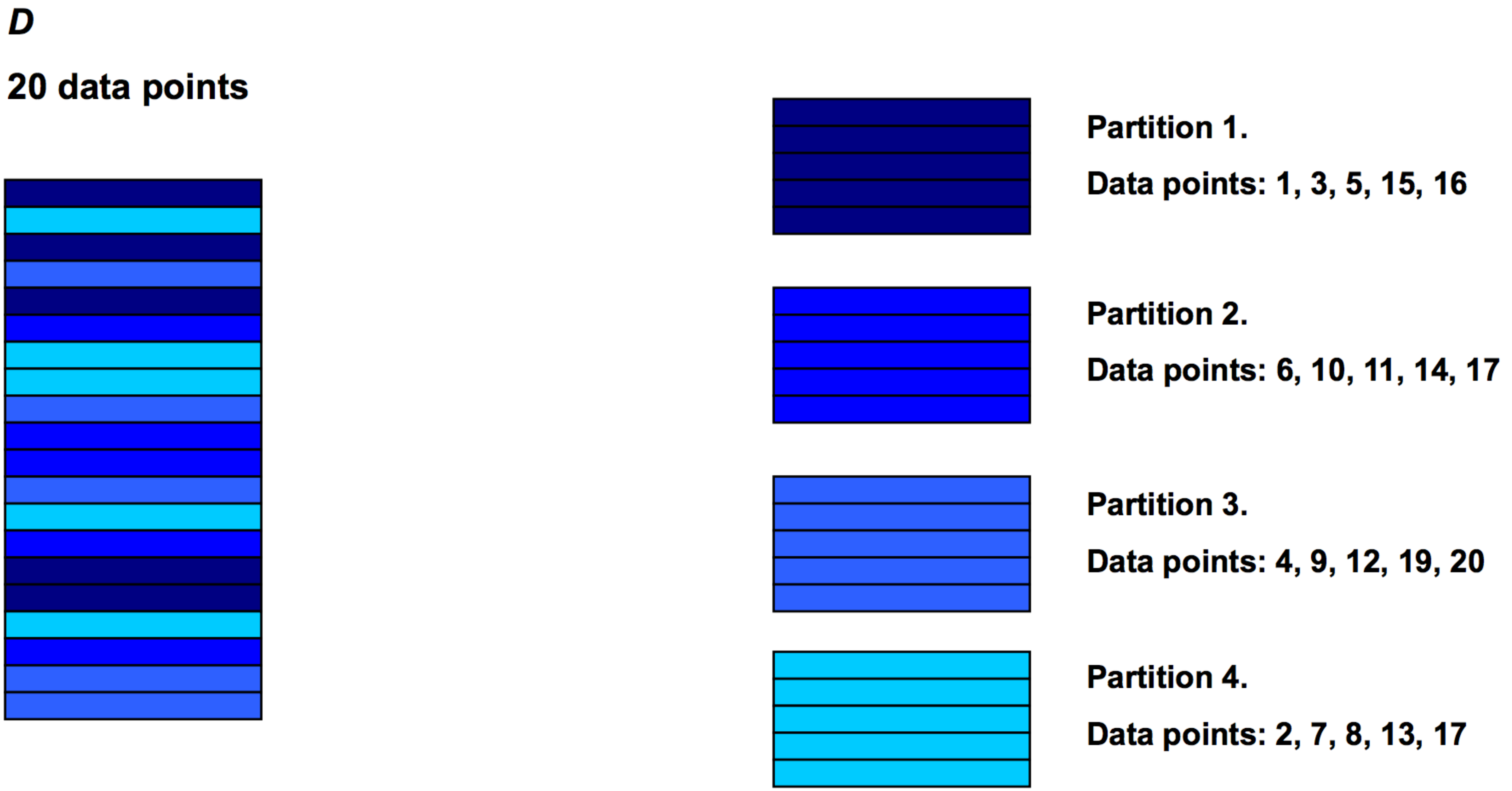
- k-fold cross validation
- Monte carlo CV
- For internal validation, common to use a single validation set or use k-fold cross validation

Monte carlo CV

- Also called “repeated learning testing-model” or repeated random subsampling
- Randomly sample without replacement the training set and the test set
    - or for smaller datasets, first sample the training set and use the rest for test
- Repeat this random subsample m times to obtain m training/test splits

k-fold CV

Randomly and evenly split into 4 non-overlapping partitions



D

20 data points

Partition 1.

Data points: 1, 3, 5, 15, 16

Partition 2.

Data points: 6, 10, 11, 14, 17

Partition 3.

Data points: 4, 9, 12, 19, 20

Partition 4.

Data points: 2, 7, 8, 13, 17

- Learn model on k-1 folds and test on the hold-out fold (done k times); average k error estimates

Bias for k-fold CV

- Train on k-1 folds, test on the other

- Each training set is only (k-1)/k as a big as the original training set; eventually, though, we will train on the entire set
  - wait, why not just remove this bias by training on only k-1 folds?

- Will this bias the estimated prediction error to be higher or lower than the true expected error?

- Bias is minimized when k = n (leave-one-out), but high-variance estimates of error (why?)

- k = 5 or 10 is an in-between that balances this bias-variance
  - (It's bias-variance all the way down)

What does this tell us about a single train-validation split?

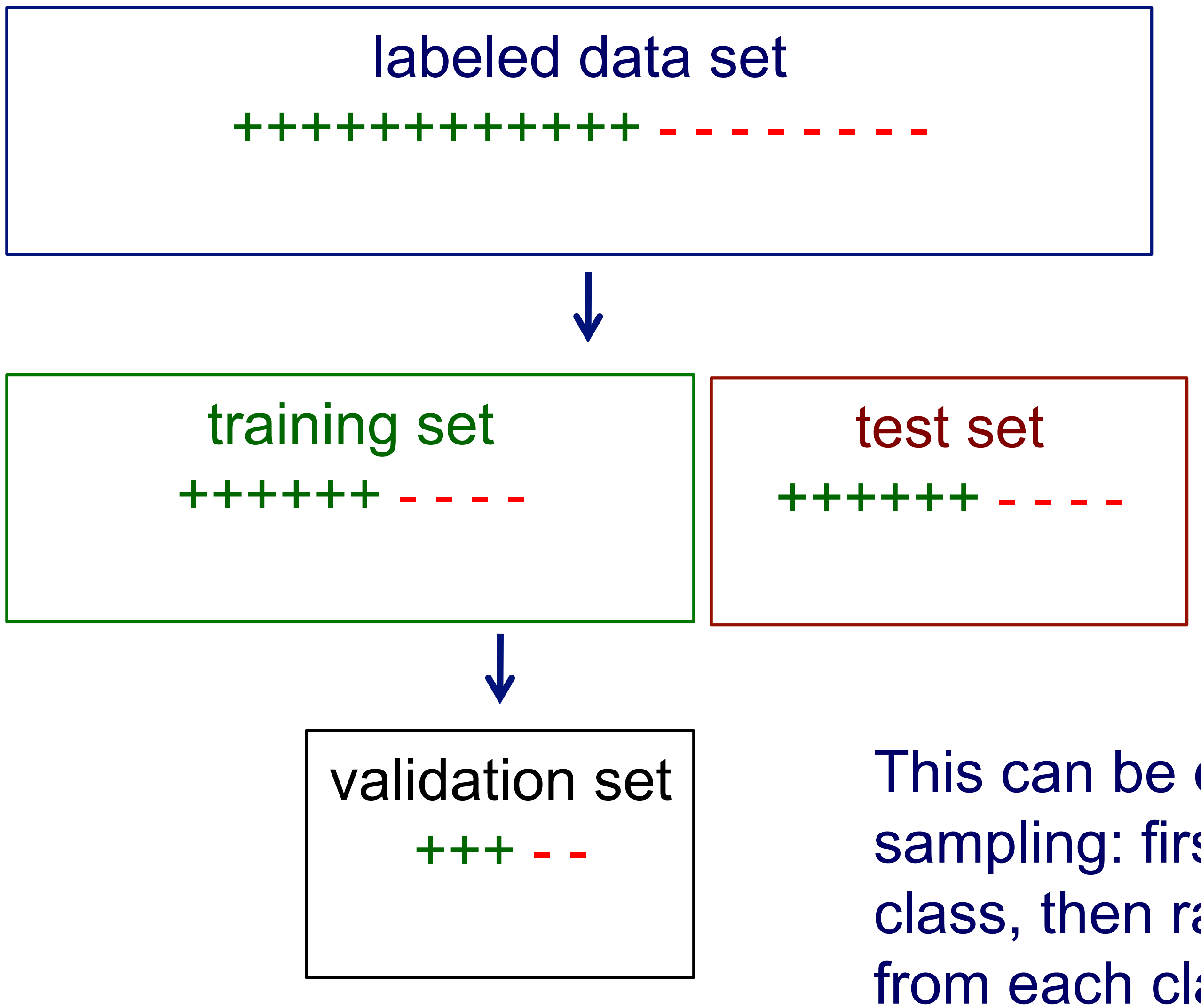
- A single split, to put aside one validation set, is almost like k-fold cross validation with k = 2
    - though, we only actually train on one fold and test on the other
- This is likely to have higher bias, and over-estimate the true expected error

LOOCV

- Why is this high-variance?
- Its too much like having one training set, and one test set
  - large correlation between k learned models
- Can overfit parameter selection to this one training set, and can find spurious connections to test set
  - find the parameters that are the best for this training set
  - if you had a different training set, maybe different parameters
  - k-fold for smaller k really does learn k models that are more significantly different, picking parameters that are “good” across training sets you could see

Stratified sampling

- When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set



This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally.

Which sampling approach should I use?

- No definitive answer, mostly empirical support
- For how to select k, bias-variance trade-off
  - for small k, high-bias and low-variance
  - for large k, low bias but high variance (e.g., leave-one-out)
  - Some experiments showing that a reasonable balance is k = 10
  - Also determined by computational resources; large k expensive
- For how to select between sampling methods,
  - repeated CV and Monte Carlo CV shown to have fewer Type 1 errors
- Criteria for internal and external CV may be quite different

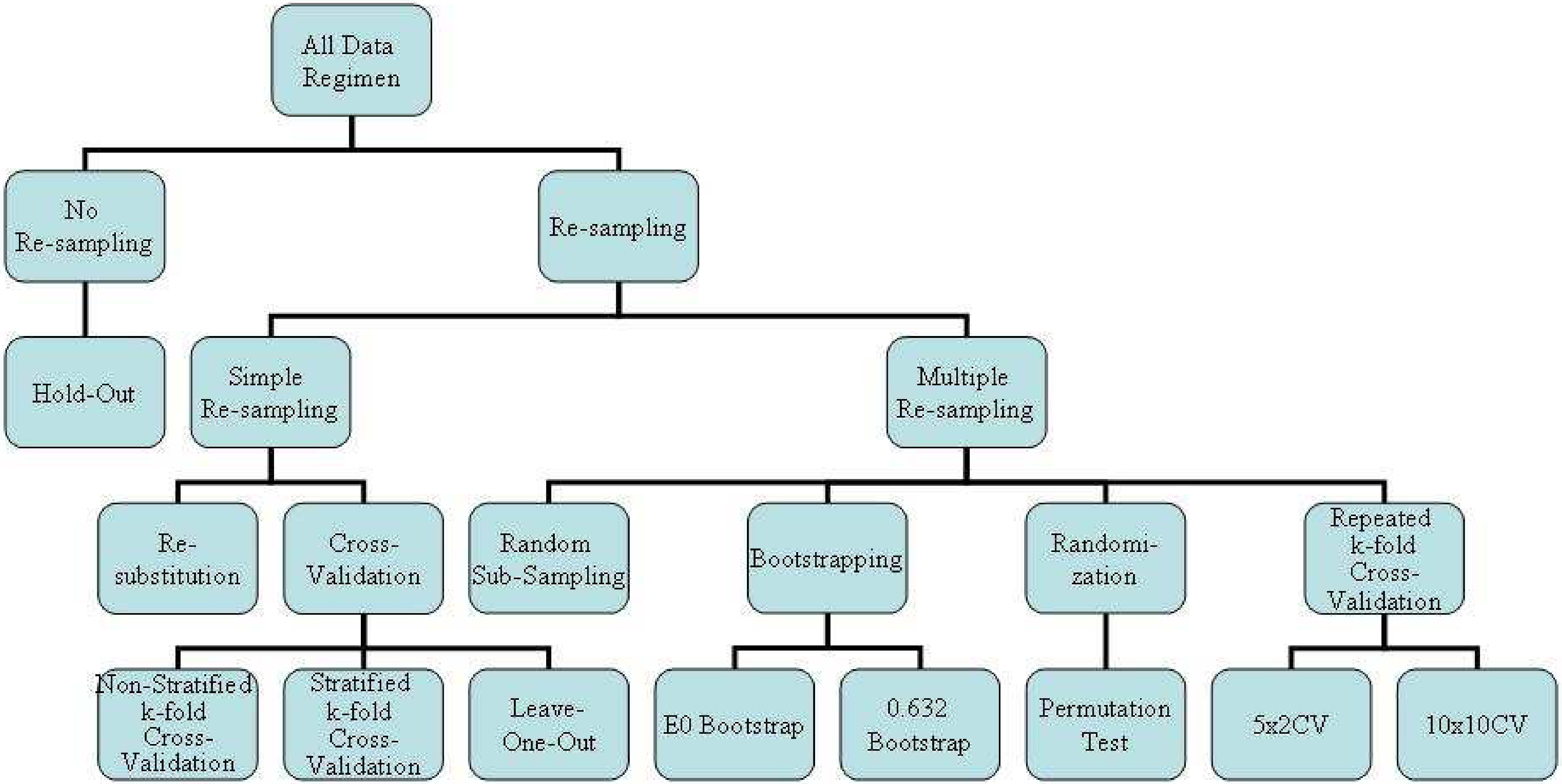
Internal

- Training k models can be expensive; want smaller k
- k-fold CV a reasonable choice because gives an almost unbiased estimate of accuracy

External

- Want to use hypothesis testing, e.g., Null Hypothesis is that the means of these two algorithms is the same
- Want a sampling technique that has less Type 1 errors
- Assumptions require independent samples of error, but empirically k-fold is not necessarily better than repeated sampling

Re-sampling summary



Experiments

- “What if I cannot find any difference between the algorithms?”
    - If you ran a fair experiment, with lots of repetitions (random splits of training and test) to get a large enough number of samples of the error, then that is a fine result
    - Remember that algorithms have many parameters that can strongly affect their performance
- “How do I select parameter ranges?”
    - the best is to provide a large enough range; this can be slow
- “Do I have to sweep all parameters in the CV?”
    - No, but remember that any choice of parameters affects your conclusion —> it is much less interesting to conclude that linear regression with regularization weight = 0.1 is outperformed by Poisson regression

Avoiding meta-parameters

- Other strategies to select meta-parameters, rather than letting data tell you the choice
- We have mathematical characterizations of generalization
- These allow some development of criteria to adjust training error
    - e.g., AIC criterion

AIC criterion

- Penalizes models with more parameters (num params = k)
  - more parameters (e.g., more hidden nodes) mean can overfit more, and of course get better likelihood on the data

$$\mathrm{AIC} = 2k - 2\ln(\mathrm{likelihood})$$

- How many parameters are there in a NN for d-dimensional inputs, 1-dimensional output, m hidden nodes?

Comparing algorithms

- Obtain samples of error/accuracy
- Reporting statistical significance
- Giving each algorithm a fair learning scenario
- Now what performance measures should we choose?

Regression objectives

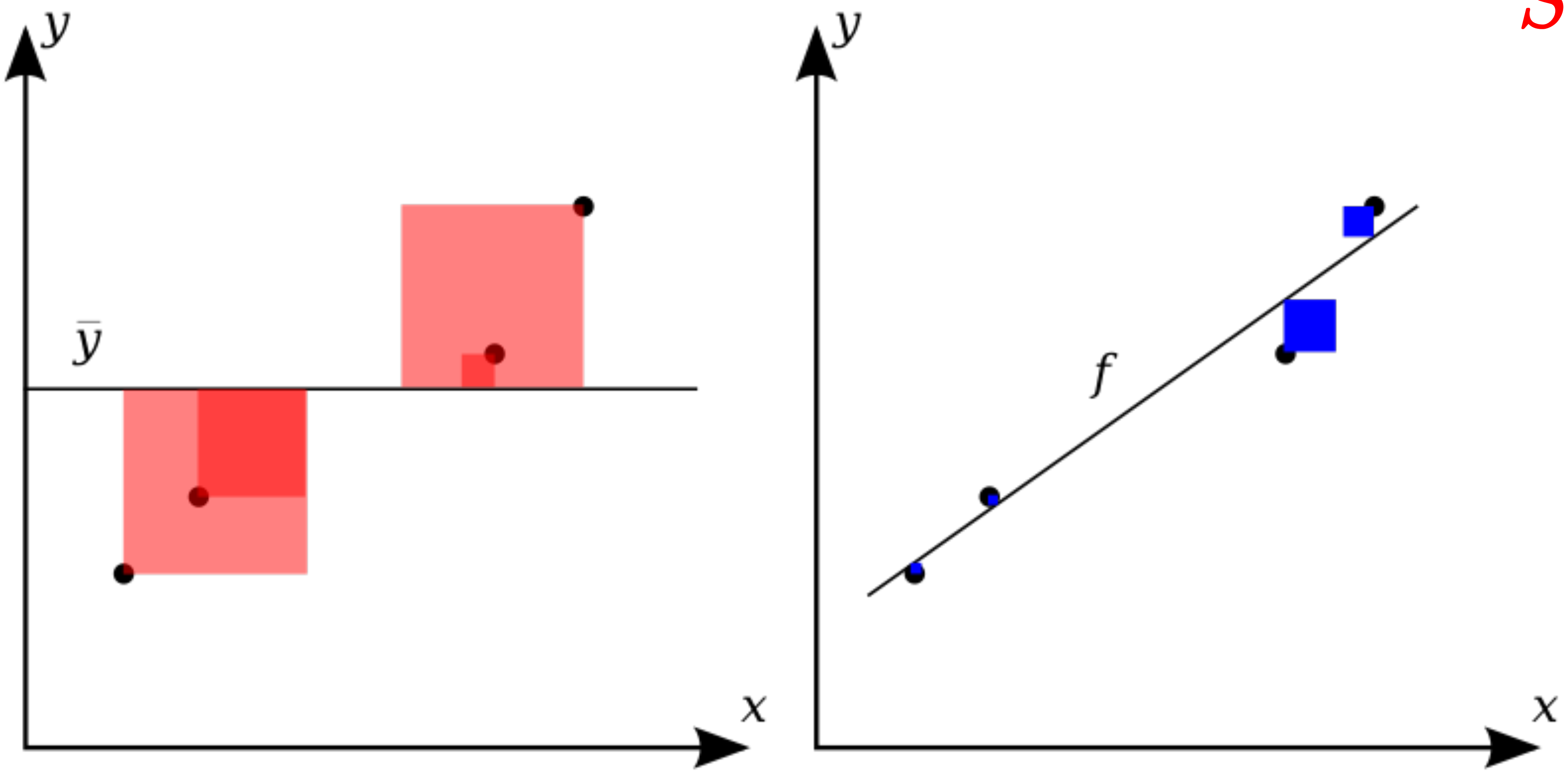
- We have looked at l2 error for estimating parameters (i.e., as an objective) and to measure performance

- Other options:
  - l1 error — can be difficult to optimize, still a useful measure of error
  - smooth l1 — smooth and convex, easier to optimize, not usually used as a measure of error (unless reporting accuracy of optimizer)
  - R-squared — coefficient of determination
  - Variance unexplained
  - Percentage error — rescale by magnitude of values

R-squared measure

- Also called “coefficient of determination”

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$



- The sum of squares of residuals, also called the residual sum of squares:

$$SS_{\text{res}} = \sum_i (y_i - f_i)^2$$

- The total sum of squares (proportional to the variance of the data):

$$SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2,$$

Larger R-squared is better

R-squared is monotone in number of features

- As add more features, the R-squared measure cannot decrease. Why?

$$R^2 \equiv 1 - \frac{SS_{\rm res}}{SS_{\rm tot}}. \qquad SS_{\rm res} = \sum_i (y_i - f_i)^2 \qquad SS_{\rm tot} = \sum_i (y_i - \bar{y})^2,$$

- Is this an issue?

- Alternative: adjusted R-squared — penalize the number of explanatory variables (features)

Percentage error

- If use error ll val1 - val2 ll, and get 0.1, is this good?
- One option: percentage errors (issues?)
- Mean absolute percentage error (MAPE)

$$\mathrm{M} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|,$$

- Symmetric MAPE

$$\mathrm{SMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

Classification terminology

- True positives — samples predicted by classifier to be positive that have true label positive

- False positives — samples predicted by classifier to be positive that have true label negative

- True negatives — samples predicted by classifier to be negative that have true label negative

- False negatives — samples predicted by classifier to be negative that have true label positive

Classification measures

Name	Symbol	Definition
Classification error	$error$	$error = \frac{fp+fn}{tp+fp+tn+fn}$
Classification accuracy	$accuracy$	$accuracy = 1 - error$
True positive rate	tpr	$tpr = \frac{tp}{tp+fn}$
False negative rate	fnr	$fnr = \frac{fn}{tp+fn}$
True negative rate	tnr	$tnr = \frac{tn}{tn+fp}$
False positive rate	fpr	$fpr = \frac{fp}{tn+fp}$
Precision	pr	$pr = \frac{tp}{tp+fp}$
Recall	rc	$rc = \frac{tp}{tp+fn}$

Why these specific values?

- These measures exist for multiple reasons
- Separate the importance of false positives and false negatives
  - In some cases, much more hazardous to have a false positive than a false negative (or vice versa)
- Avoid issues with imbalanced datasets

Confusion Matrix for binary classification

True class

Predicted class

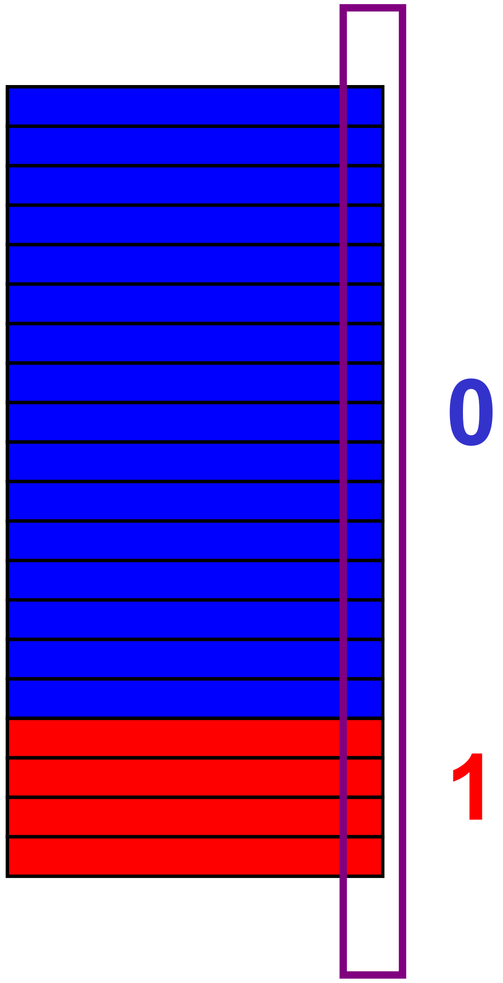
	0	**1**
0	☺ N_{00} tn	☹ N_{01} fn
1	☹ fp N_{10}	☺ N_{11} tp

Number of data points whose true class was 0 but predicted class was 1.

$$Accuracy = \frac{N_{00} + N_{11}}{N_{00} + N_{10} + N_{01} + N_{11}}$$

$$Error = 1 - Accuracy$$

Example of importance of measures: imbalanced datasets



16 data points have class 0 (majority class)

4 data points have class 1 (minority class)

Trivial classifier: always predict majority class

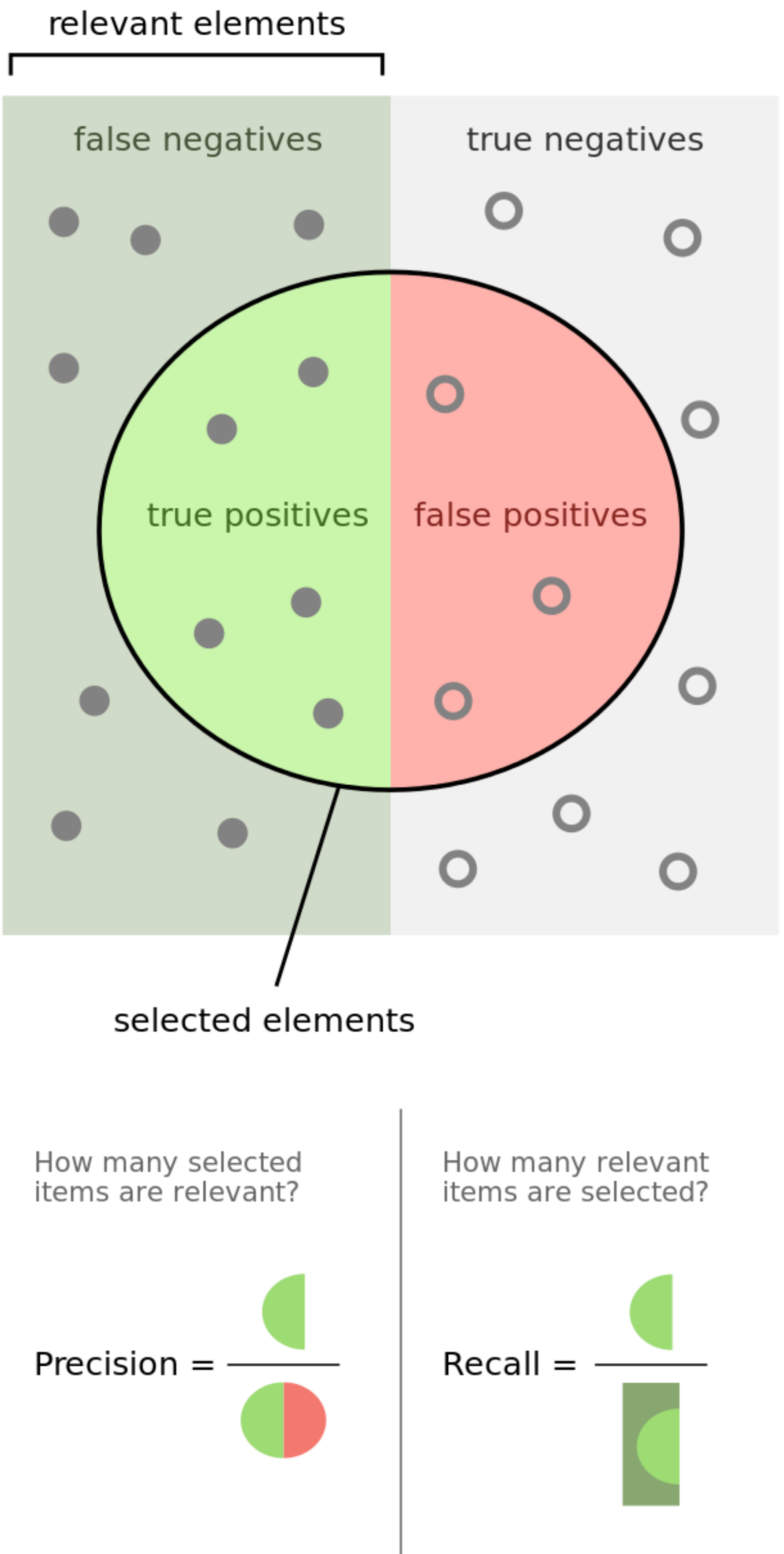
Accuracy of a trivial classifier is: 16/20 = 80%

Precision and recall

- Example: when a search engine returns 30 pages only 20 of which were relevant while failing to return 40 additional relevant pages, its precision is 20/30 = 2/3 while its recall is 20/60 = 1/3.

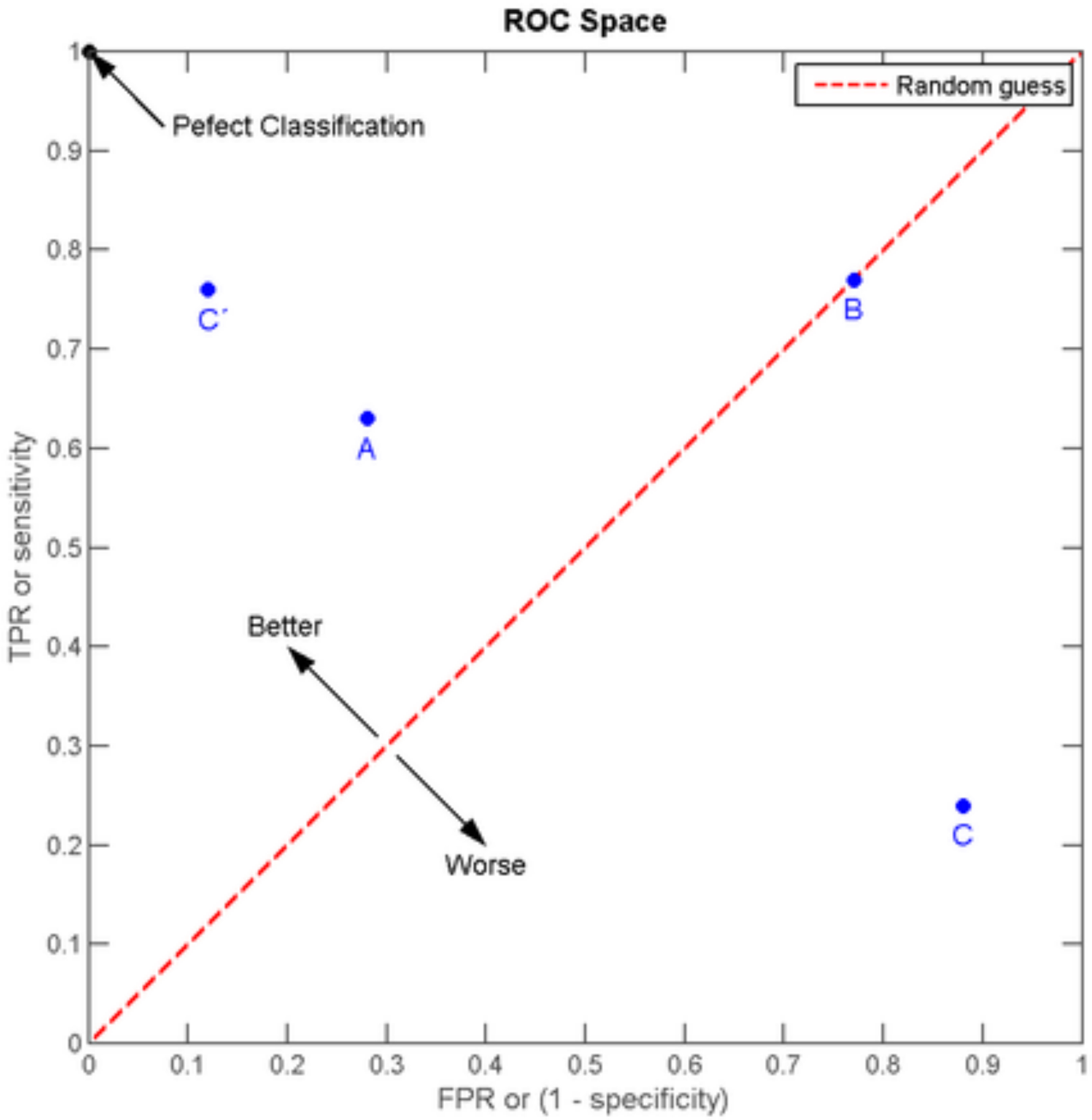
$$\text{recall} = \frac{\text{tp}}{\text{fn} + \text{tp}}$$

$$\text{precision} = \frac{\text{tp}}{\text{fp} + \text{tp}}$$



ROC space

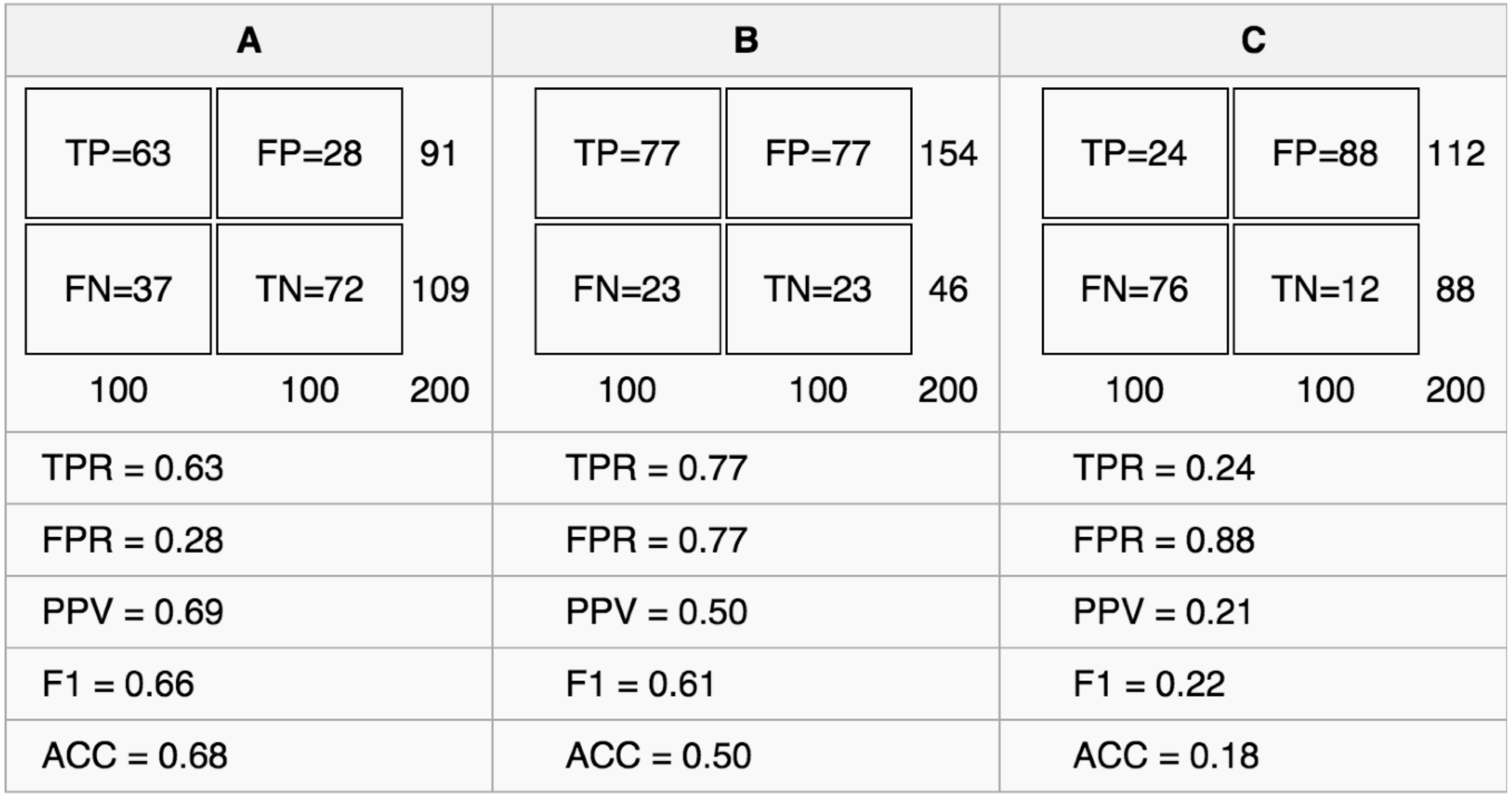
Predict positive if p(y=1|x) > threshold



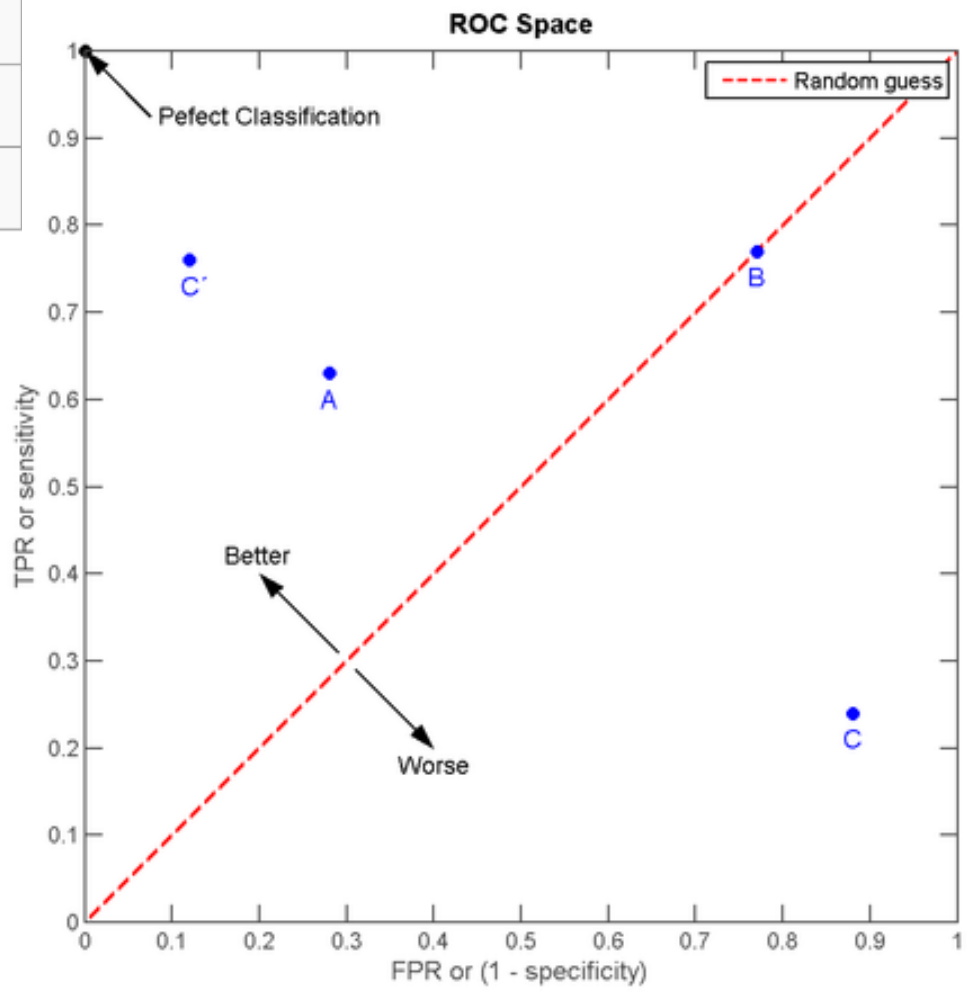
Threshold = 0

Threshold = 1

ROC space

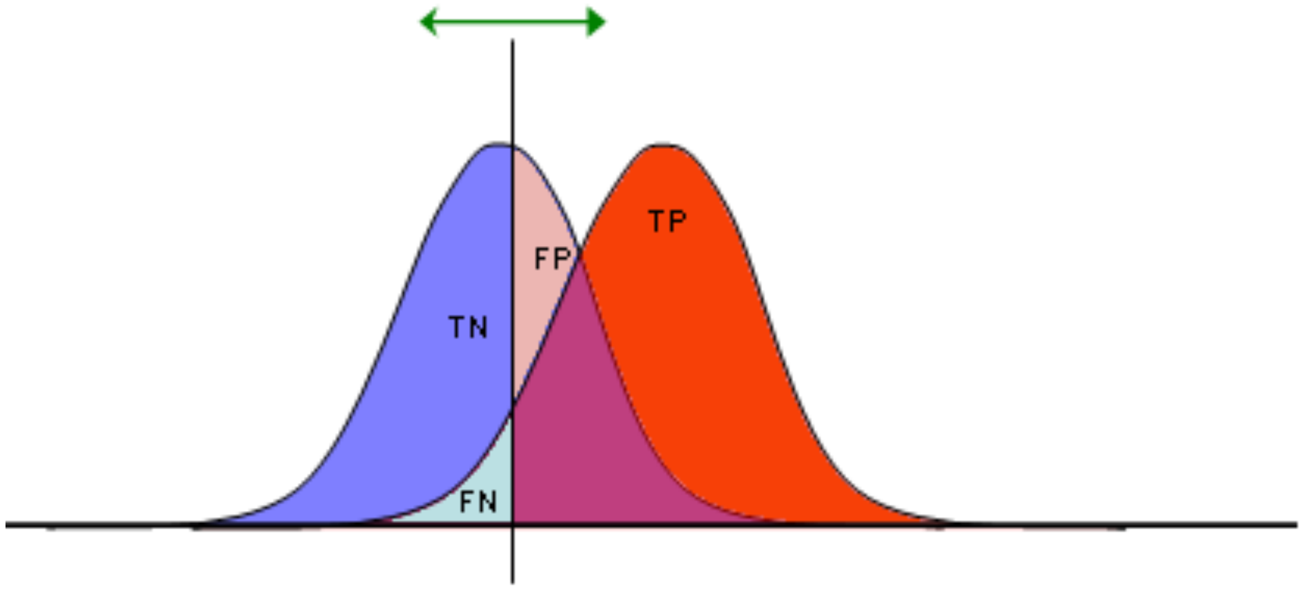


A			B			C		
TP=63	FP=28	91	TP=77	FP=77	154	TP=24	FP=88	112
FN=37	TN=72	109	FN=23	TN=23	46	FN=76	TN=12	88
100	100	200	100	100	200	100	100	200
TPR = 0.63			TPR = 0.77			TPR = 0.24		
FPR = 0.28			FPR = 0.77			FPR = 0.88		
PPV = 0.69			PPV = 0.50			PPV = 0.21		
F1 = 0.66			F1 = 0.61			F1 = 0.22		
ACC = 0.68			ACC = 0.50			ACC = 0.18		

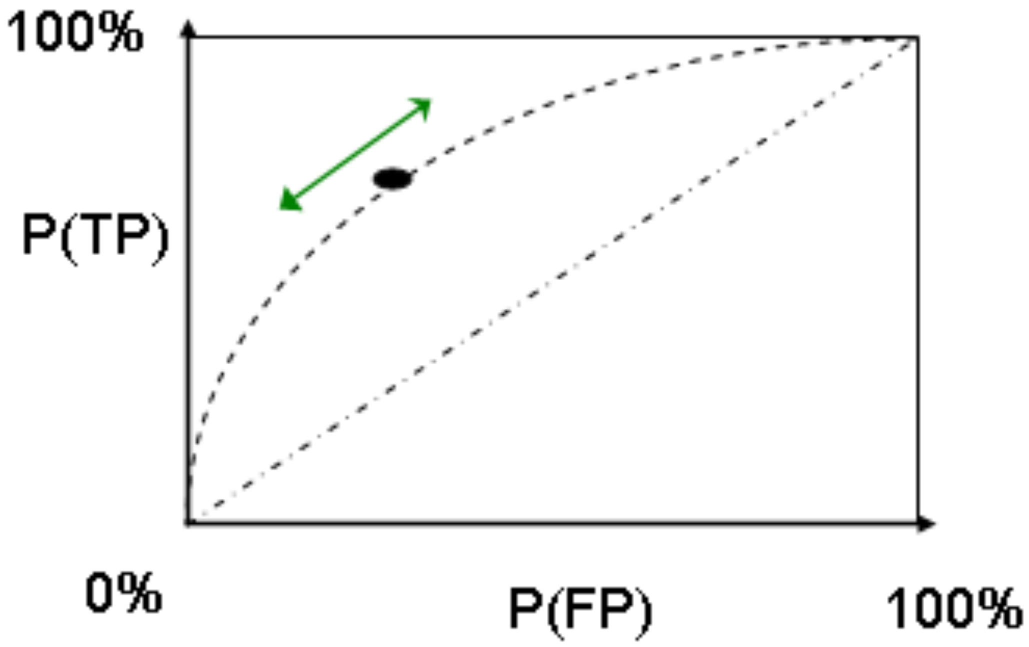


ROC Curve example

e.g., diseased people, healthy people
blood protein levels normally distributed
Parameter that changes: threshold

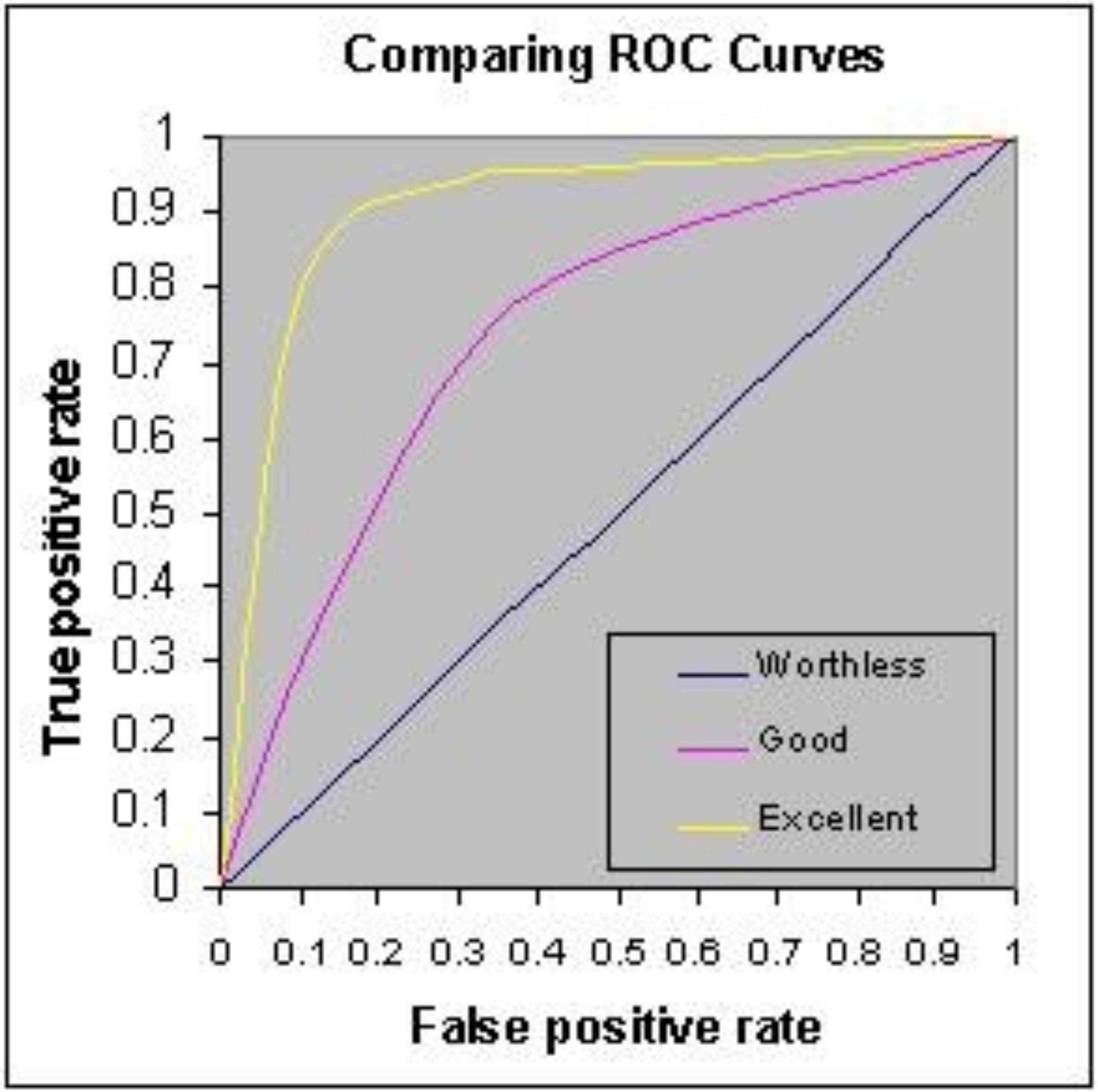


TP	FP
FN	TN
1	1



ROC Curve

Predict positive if
$p(y=1|x) >$ threshold



Threshold = 0

Threshold = 1

Area under the curve

- AUC or AUCROC gives the area under the ROC curve
- AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one
- Some issues in using AUC to compare classifiers (see “Measuring classifier performance: a coherent alternative to the area under the ROC curve”, Hand, JMLR, 2009)
  - can give unequal important to a FPR or TPR for different classifiers

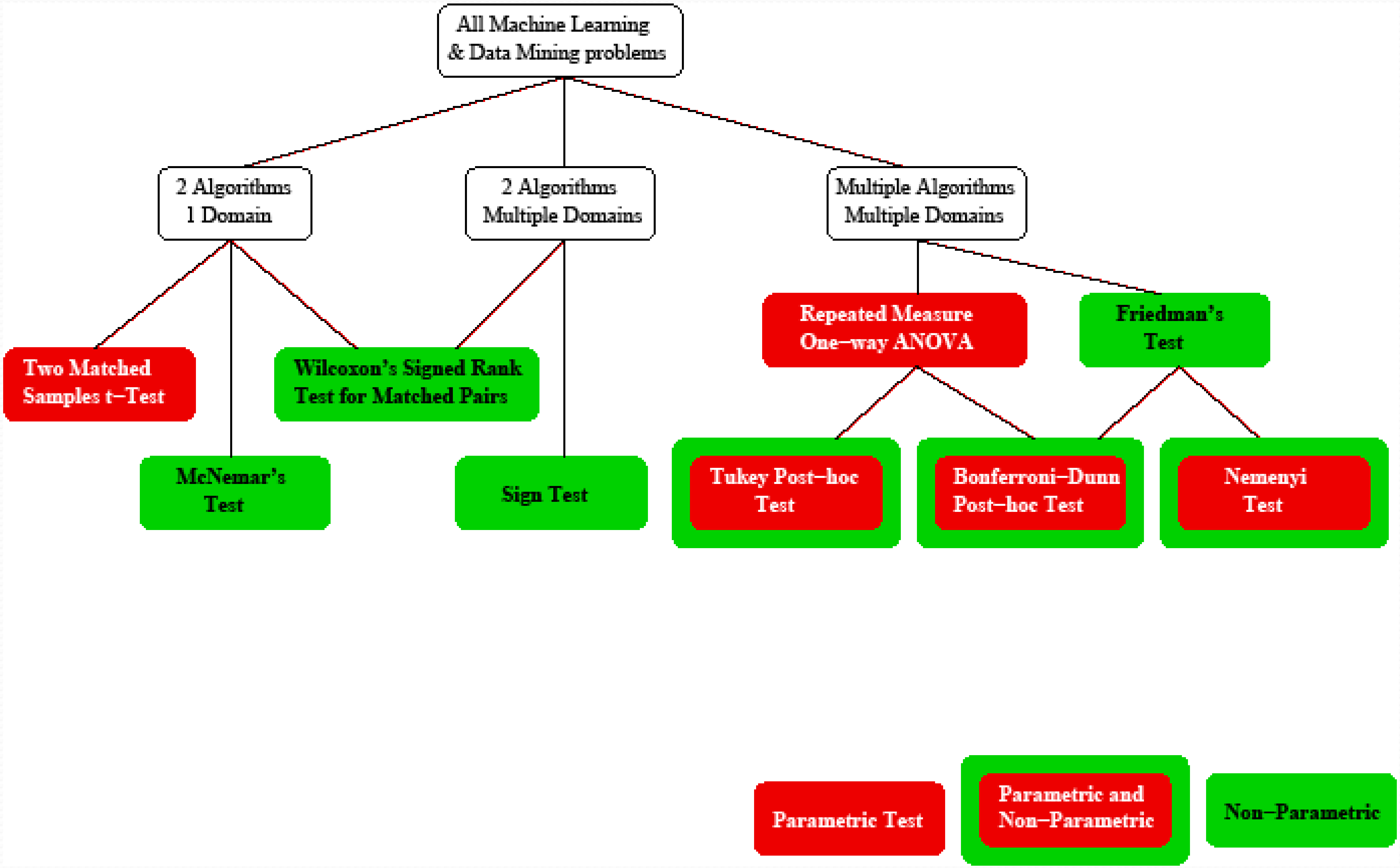
Whiteboard

- Statistical significance tests
- Rademacher complexity

How to choose tests?

- Try to satisfy assumptions and use some rules of thumb
- Parametric statistical tests make stronger assumptions about the distribution of the data
- Non-parametric tests make weaker assumptions, but are less powerful (less able to reject the null hypothesis when it is false)
- Selection based on type of problem
  - comparing 2 algorithms on a single domain
  - comparing 2 algorithms across domains
  - comparing multiple algorithms across domains

Statistical test summary



All Machine Learning & Data Mining problems

2 Algorithms 1 Domain

2 Algorithms Multiple Domains

Multiple Algorithms Multiple Domains

Two Matched Samples t–Test

McNemar's Test

Wilcoxon's Signed Rank Test for Matched Pairs

Sign Test

Repeated Measure One–way ANOVA

Friedman's Test

Tukey Post–hoc Test

Bonferroni–Dunn Post–hoc Test

Nemenyi Test

Parametric Test

Parametric and Non–Parametric

Non–Parametric

Assumptions of the t-test

- The Normality or Pseudo-Normality Assumption: samples come from normally distributed population. Alternatively, the sample size of the testing set should be greater than 30.

- The Randomness of the Samples: The sample should be representative of the underlying population. Therefore, the instances of the testing set should be randomly chosen from their underlying distribution.

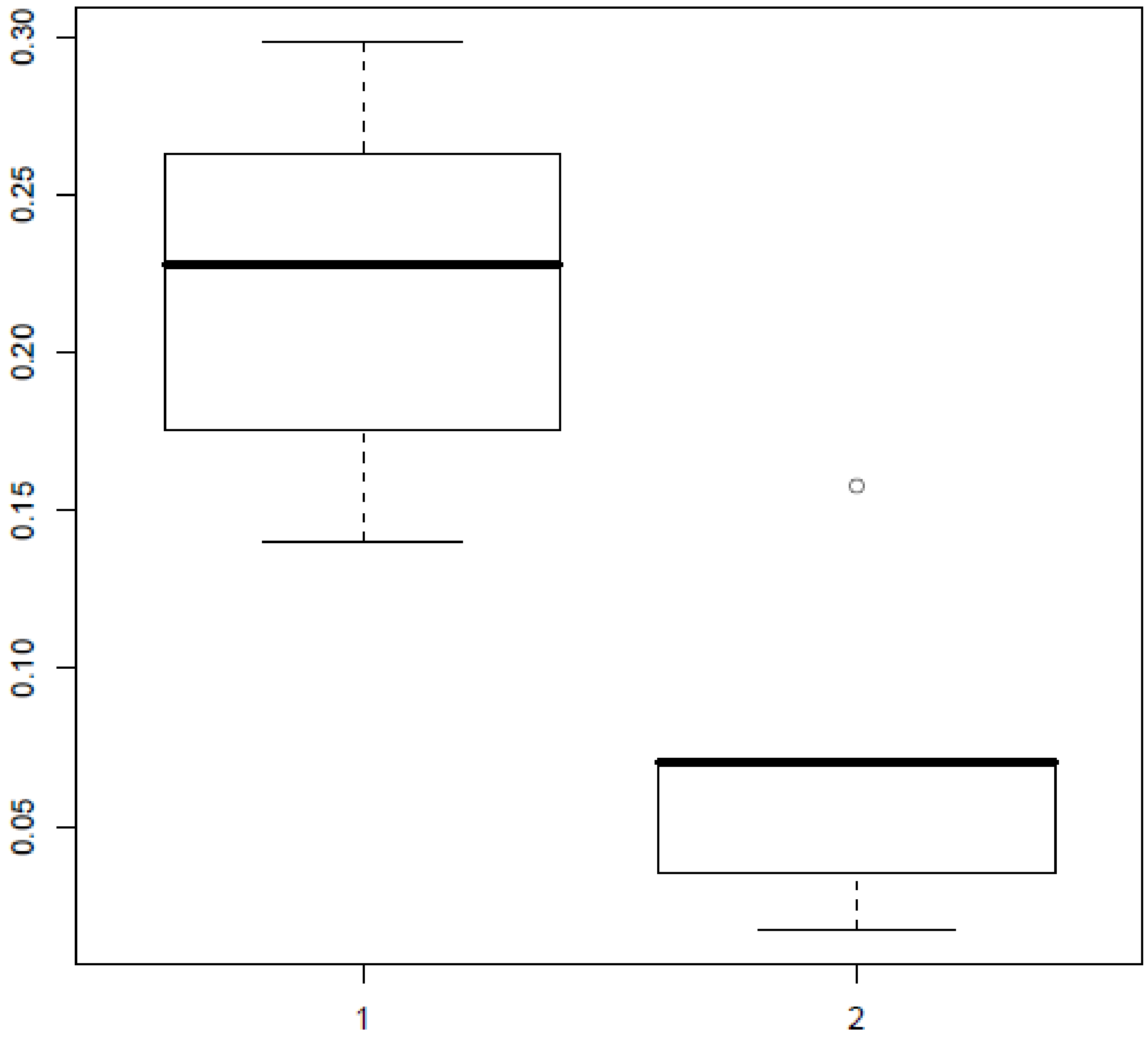
- Equal Variance of the populations: The two samples come from populations with equal variance.

Example where assumptions of t-test violated

- Equal Variance: variance of C4.5 and NB cannot be considered equal.

- Not warranted to use the t-test to compare C4.5 to NB on the Labour data.

- A better test to use is
  - Welch’s t-test
  - non-parametric alternative, McNemar’s Test



One-tailed versus two-tailed

- One-sided question: is algorithm 1 better than algorithm 2?
- Two-sided question: is algorithm 1 and 2 two different?
  - i.e., either could be better
- Usually we care about one-sided
  - p = Pr (T > t), where T is a random variable
  - for paired t-test, little t reflects the average difference scaled by variance and samples
  - t = average difference / sqrt(standard deviation x numsamples)