Evaluation of Machine Learning Classifiers

Machine Learning

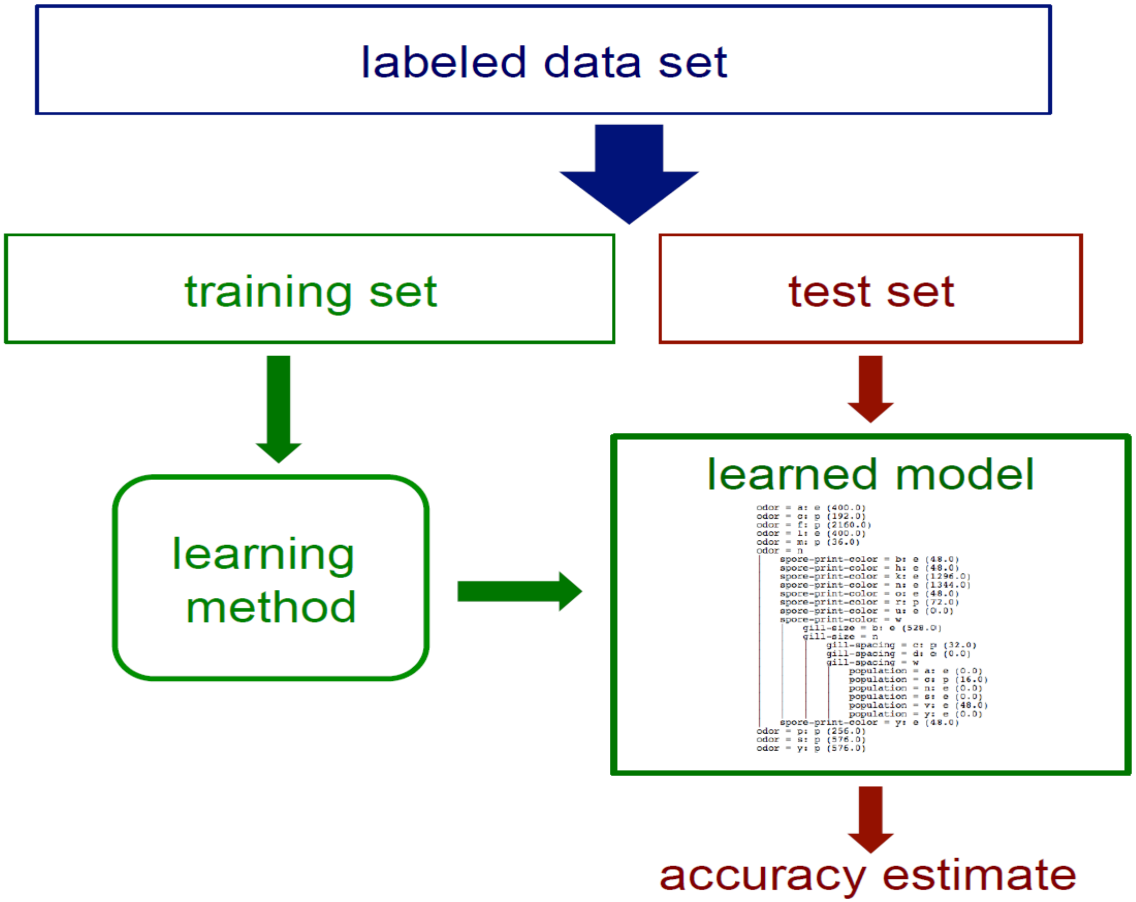
Dr. Dinesh K. Vishwakarma

Outline: Evaluation Parameters

- Precision
- Recall
- Accuracy
- F-Measure
- True Positive Rate
- False Positive Rate
- Sensitivity
- ROC

Experiment: Training and Testing

- Objective: Unbiased estimate of accuracy

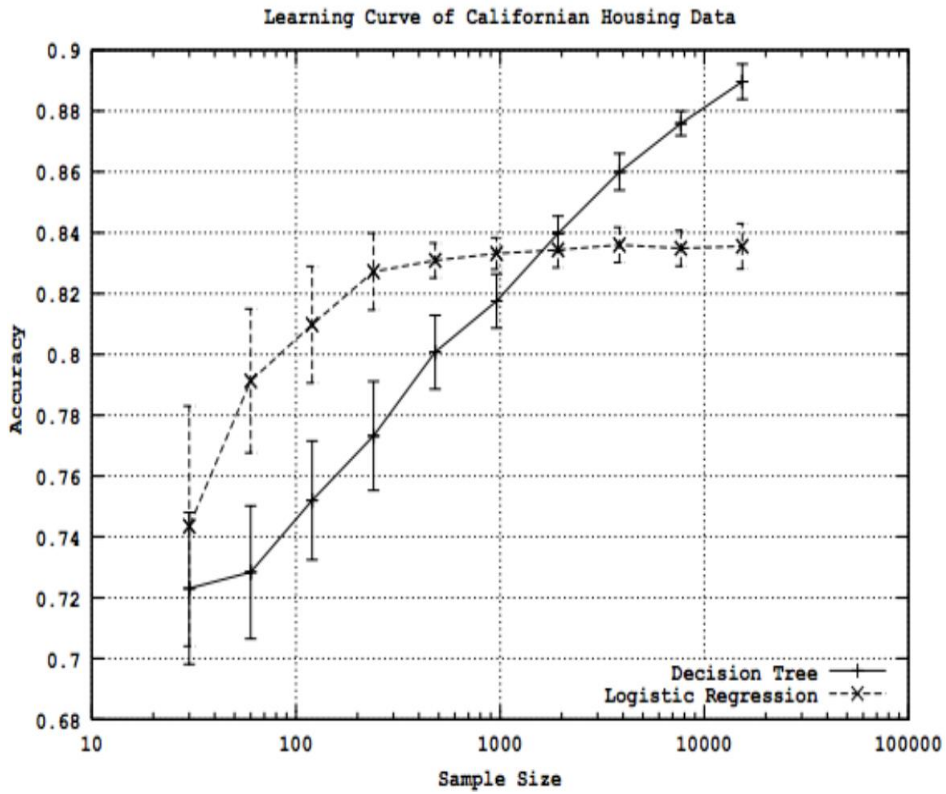


Experiment: Training and Testing…

- How can we get an unbiased estimate of the accuracy of a learned model?
  - ✓ when learning a model, you should pretend that you don’t have the test data yet (it is “in the mail”)*
  - ✓ if the test-set labels influence the learned model in any way, accuracy estimates will be biased
- * In some applications it is reasonable to assume that you have access to the feature vector (i.e. ***x***) but not the ***y*** part of each test instance

Learning Curve

- How does the accuracy of a learning method change as a function of the training-set size?
  - ✓ This can be assessed by plotting *learning curves*

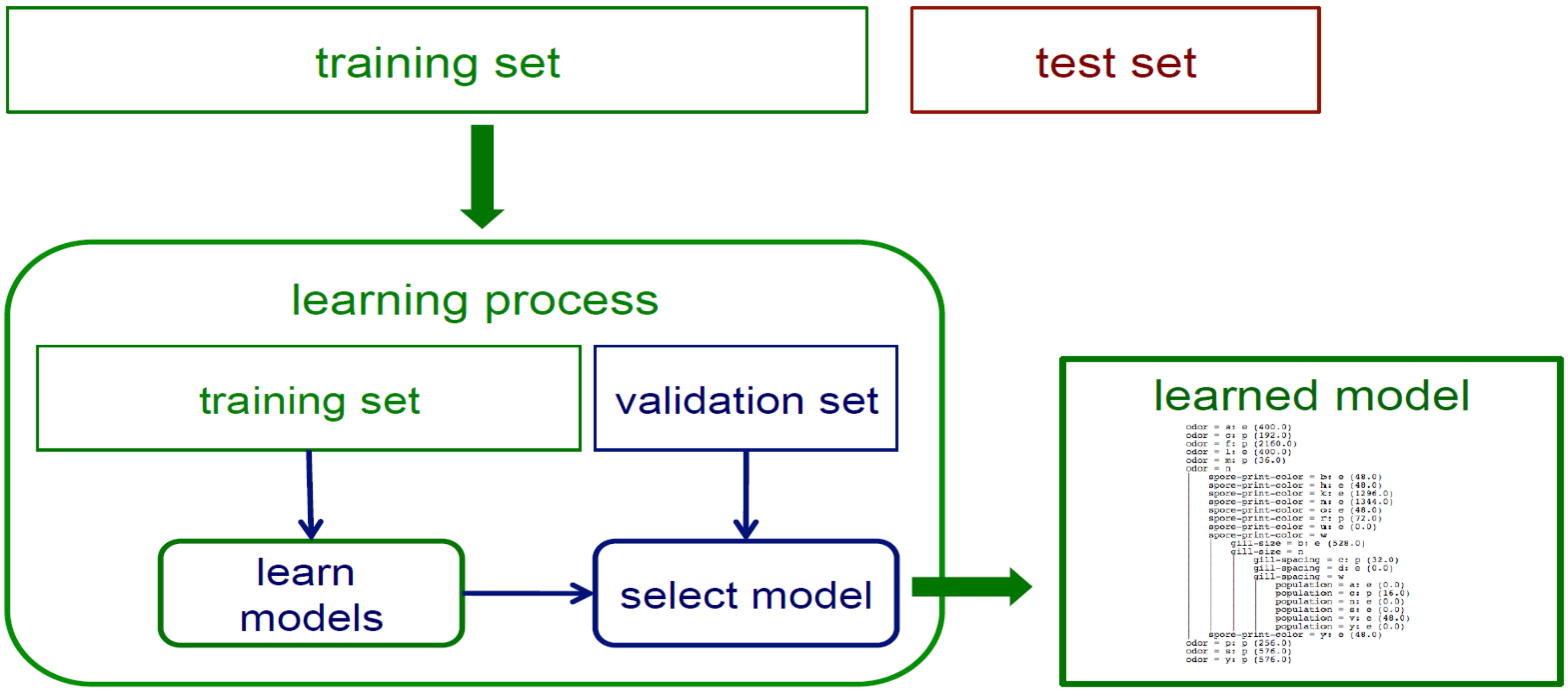


#Given training/test set partition

- for each sample size *s* on learning curve
- (optionally) repeat *n* times
- randomly select *s* instances from training set
  - learn model
- evaluate model on test set to determine accuracy *a*
- plot (*s*, *a*) or (*s*, avg. accuracy and error bars)

Validation (Tuning) Set

- Consider we want unbiased estimates of accuracy during the learning process (e.g. to choose the best level of decision-tree pruning)?



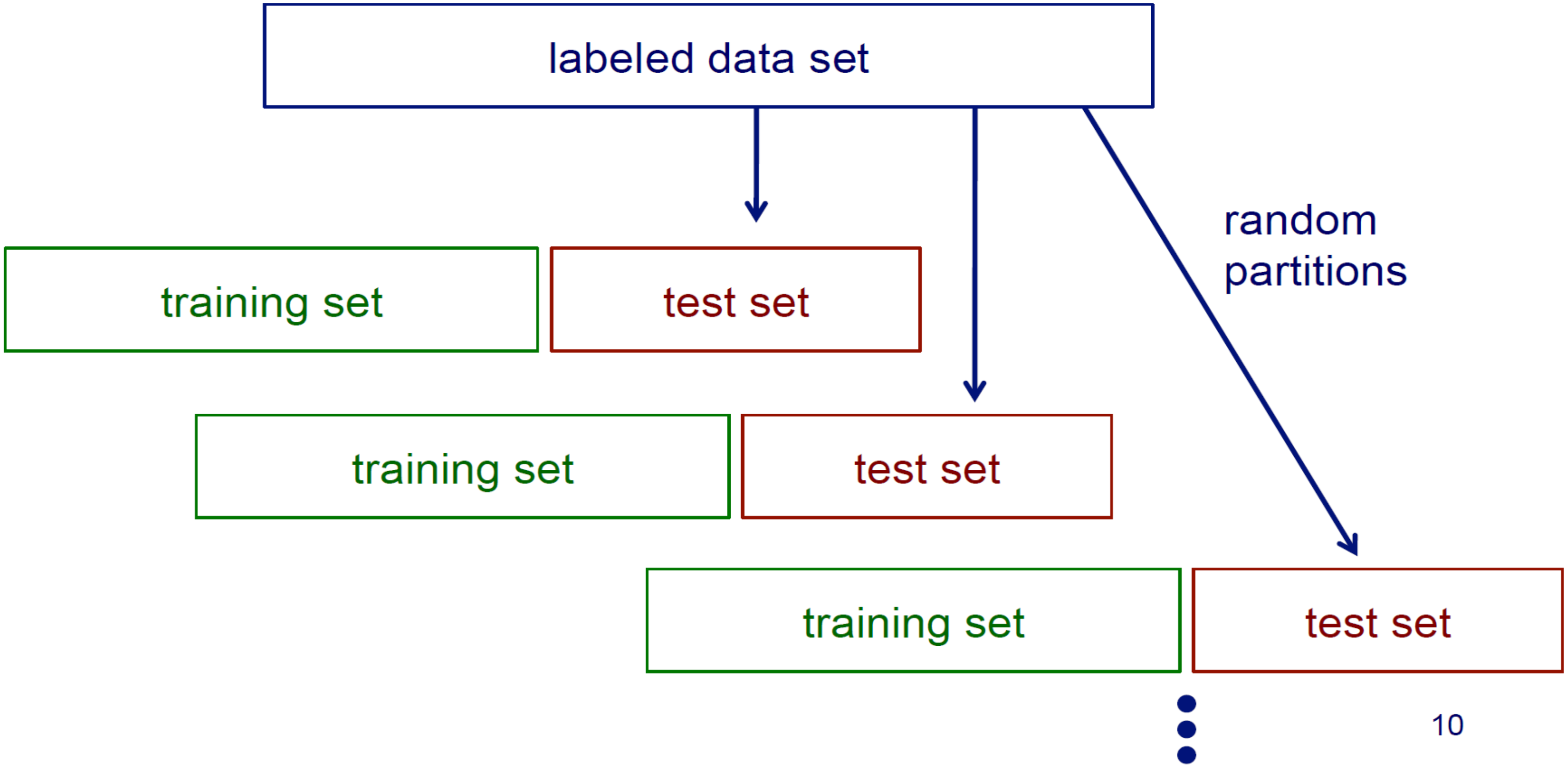
Partition training data into separate training/validation sets

Limitation of Single Training/Test Partition

- We may not have enough data to make sufficiently large
  - ✓ training and test sets a larger test set gives us more reliable estimate of accuracy (i.e. a lower variance estimate)
  - ✓ but… a larger training set will be more representative of how much data we actually have for learning process
- A single training set doesn't tell us how sensitive accuracy is to a particular training sample

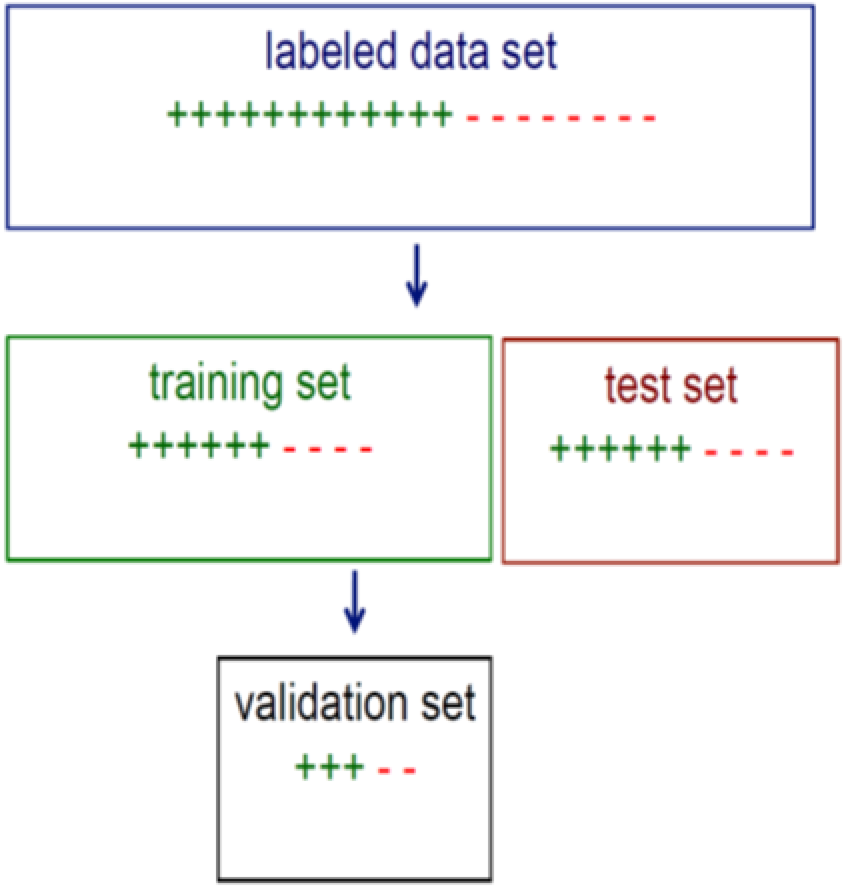
Random Sampling

- It can be addressed the second issue by repeatedly randomly partitioning the available data into training and set sets.



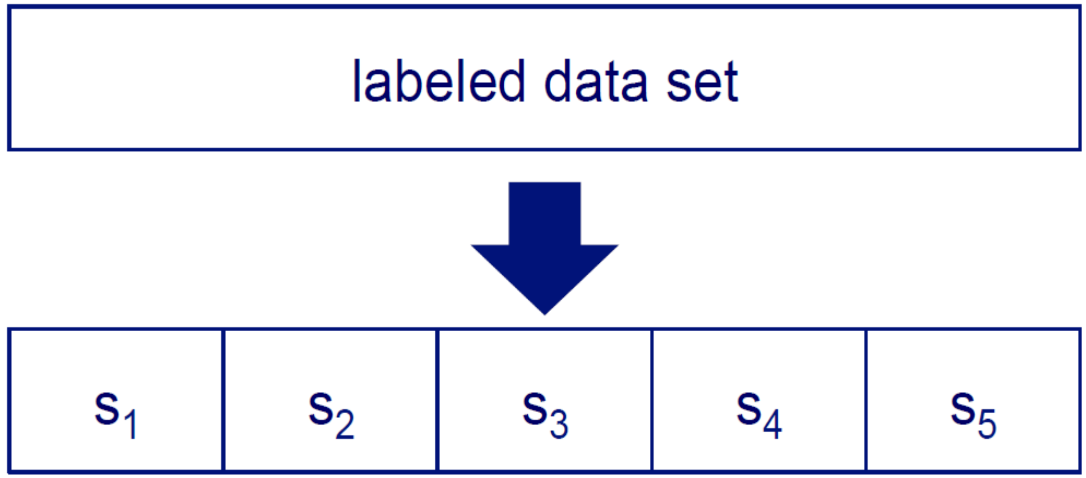
Random Sampling…

- 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.



Cross Validation

Partition data into *n* subsamples



Iteratively leave one subsample out for the test set, train on the rest

iteration	train on	test on
1	s_2 s_3 s_4 s_5	s_1
2	s_1 s_3 s_4 s_5	s_2
3	s_1 s_2 s_4 s_5	s_3
4	s_1 s_2 s_3 s_5	s_4
5	s_1 s_2 s_3 s_4	s_5

Cross Validation Example

- Suppose we have 100 instances, and we want to estimate accuracy with cross validation.

iteration	train on	test on	correct
1	s_2 s_3 s_4 s_5	s_1	11 / 20
2	s_1 s_3 s_4 s_5	s_2	17 / 20
3	s_1 s_2 s_4 s_5	s_3	16 / 20
4	s_1 s_2 s_3 s_5	s_4	13 / 20
5	s_1 s_2 s_3 s_4	s_5	16 / 20

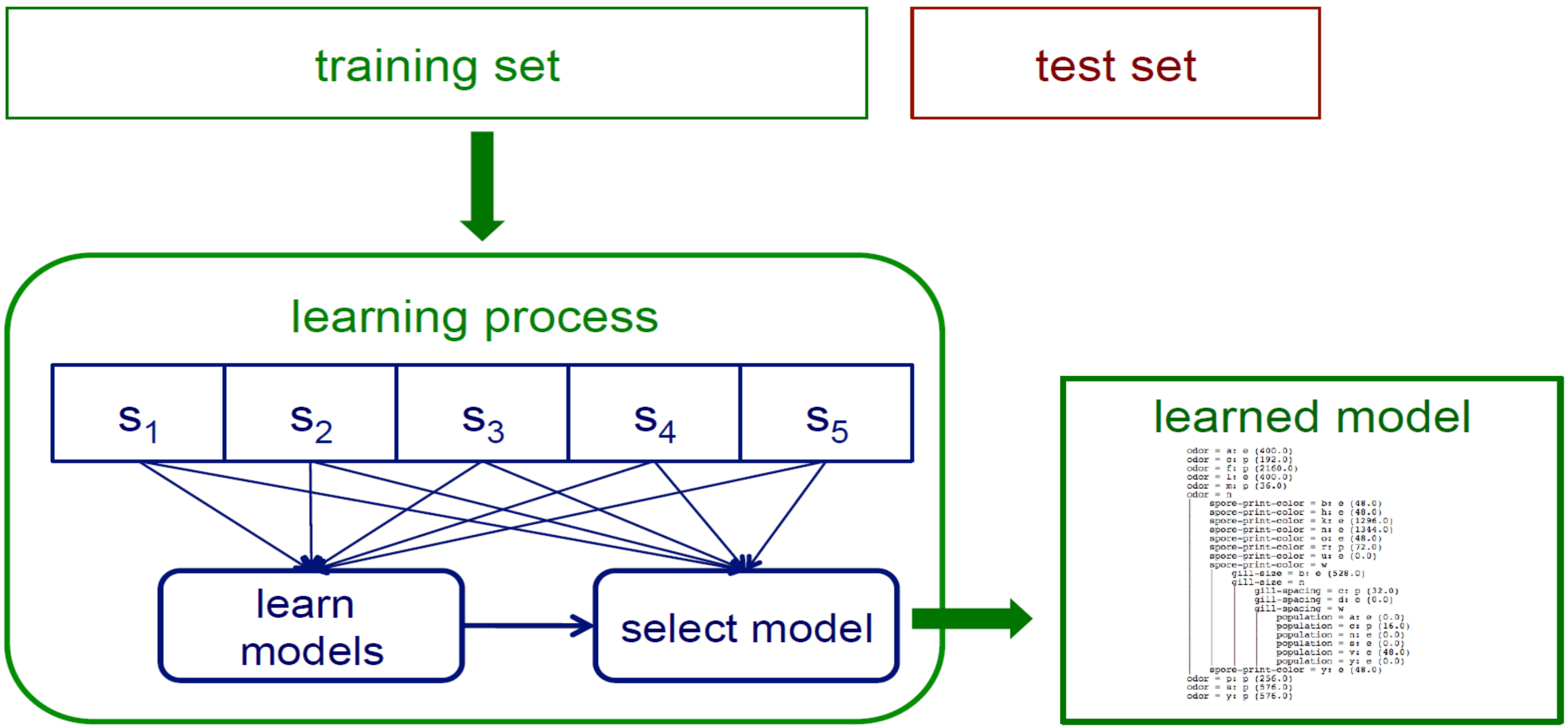
accuracy = 73/100 = 73%

Cross Validation…

- 10-fold cross validation is common, but smaller values of *n* are often used when learning takes a lot of time
- In *leave-one-out* cross validation, *n* = # instances
- In *stratified* cross validation, stratified sampling is used when partitioning the data
- CV makes efficient use of the available data for testing
- Note that whenever we use multiple training sets, as in CV and random resampling, we are evaluating a learning method as opposed to an individual learned model

Internal Cross Validation

- Instead of a single validation set, we can use cross-validation within a training set to select a model (e.g. to choose the best level of decision-tree pruning)



Example: using internal cross validation to select *k* in *k*-NN

- Given a training set
  1. partition training set into *n* folds, $s_1 \ldots \ldots \ldots s_n$
  2. for each value of $\boldsymbol{k}$ considered

     for $i = 1\ to\ n$

     learn *k*-NN model using all folds but *si*

     evaluate accuracy on s_i
  3. select *k* that resulted in best accuracy for $s_1 \ldots \ldots \ldots s_n$
  4. learn model using entire training set and selected $\boldsymbol{k}$.
- The steps inside the box are run independently for each training set (i.e. if we're using 10-fold CV to measure the overall accuracy of our *k*-NN approach, then the box would be executed 10 times)

Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision $P = \frac{t_p}{t_p + f_p}$
- Recall $\mathrm{R} = \frac{t_p}{t_p + f_n}$

Issues with “Precision & Recall”

True class →	Pos	Neg
Yes	200 TP	100 FP
No	300 FN	400 TN
	P=500	N=500

True class →	Pos	Neg
Yes	200	100
No	300	0
	P=500	N=100

- Both classifiers gives the ***same*** *precision* and *recall* values of 66.7% and 40% (Note: the data sets are different)
- They exhibit very different behaviours:
  - ✓ Same **positive** recognition rate
  - ✓ Extremely different ***negative*** recognition rate: strong on the left / nil on the right
- Note: Accuracy has no problem catching this!

A combined measure: *F*

- Combined measure that assesses **precision/recall** tradeoff is **F measure** (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- People usually use balanced F_1 measure
  - i.e., with $\beta = 1$ or $\alpha = ½$
- Harmonic mean is a conservative average.

Accuracy Measure

- Given a query, an engine classifies each doc as "Relevant" or "Nonrelevant"
- The **accuracy** of an engine: the fraction of these classifications that are correct
  - $Accuracy(\%) = \frac{(t_p + t_n)}{(t_p + t_n + f_n + f_p)} \times 100$
- **Accuracy** is a commonly used for evaluation measure in machine learning.

Issues with Accuracy

- **Consider a 2-class problem**
  - *Number of Class 0 examples = 9990*
  - *Number of Class 1 examples = 10*
- **If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %**
  - *Accuracy is misleading because model does not detect any class 1 example*

Issues with Accuracy…

True class →	Pos	Neg
Yes	200	100
No	300	400
	P=500	N=500

True class →	Pos	Neg
Yes	400	300
No	100	200
	P=500	N=500

- Both classifiers gives 60% accuracy.
- They exhibit very different behaviours:
  - ✓ **On the left:** weak positive recognition rate/strong negative recognition rate
  - ✓ **On the right:** strong positive recognition rate/weak negative recognition rate

Is accuracy adequate measure?

- **Accuracy may not be useful measure in cases where**
  - there is a large class skew
    - ✓ Is 98% accuracy good if 97% of the instances are negative?
  - there are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong.
    - ✓ Consider a medical domain in which a false positive results in an extraneous test but a false negative results in a failure to treat a disease
  - we are most interested in a subset of high-confidence predictions

Miss Classification Error

- Recognition rate=accuracy=success rate

Hypothesized class (prediction)		Classified +ve	Classified –ve
Actual class (observation)	Actual +ve	TP	FN
	Actual –ve	FP	TN

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: [illegible]	[illegible]	[illegible]

Table 2.2 A confusion matrix of a model

	Predicted +1	**Predicted –1**
Actual +1	95	7
Actual –1	4	94

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}};$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}};$$

Other form of Accuracy Metrics

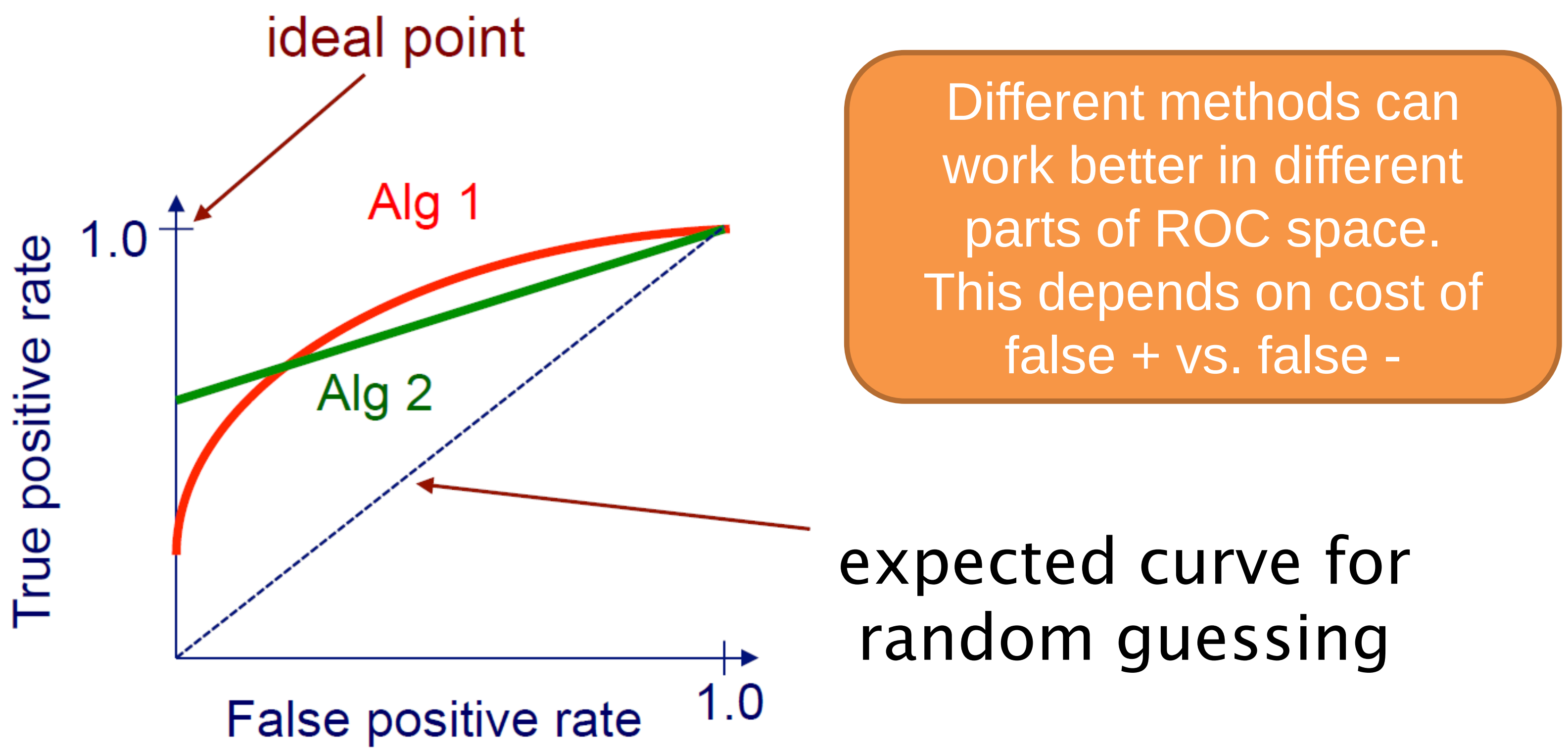
predicted class		actual class: positive	actual class: negative
	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

$$\text{true positive rate (recall)} = \frac{\mathrm{TP}}{\text{actual pos}} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$$

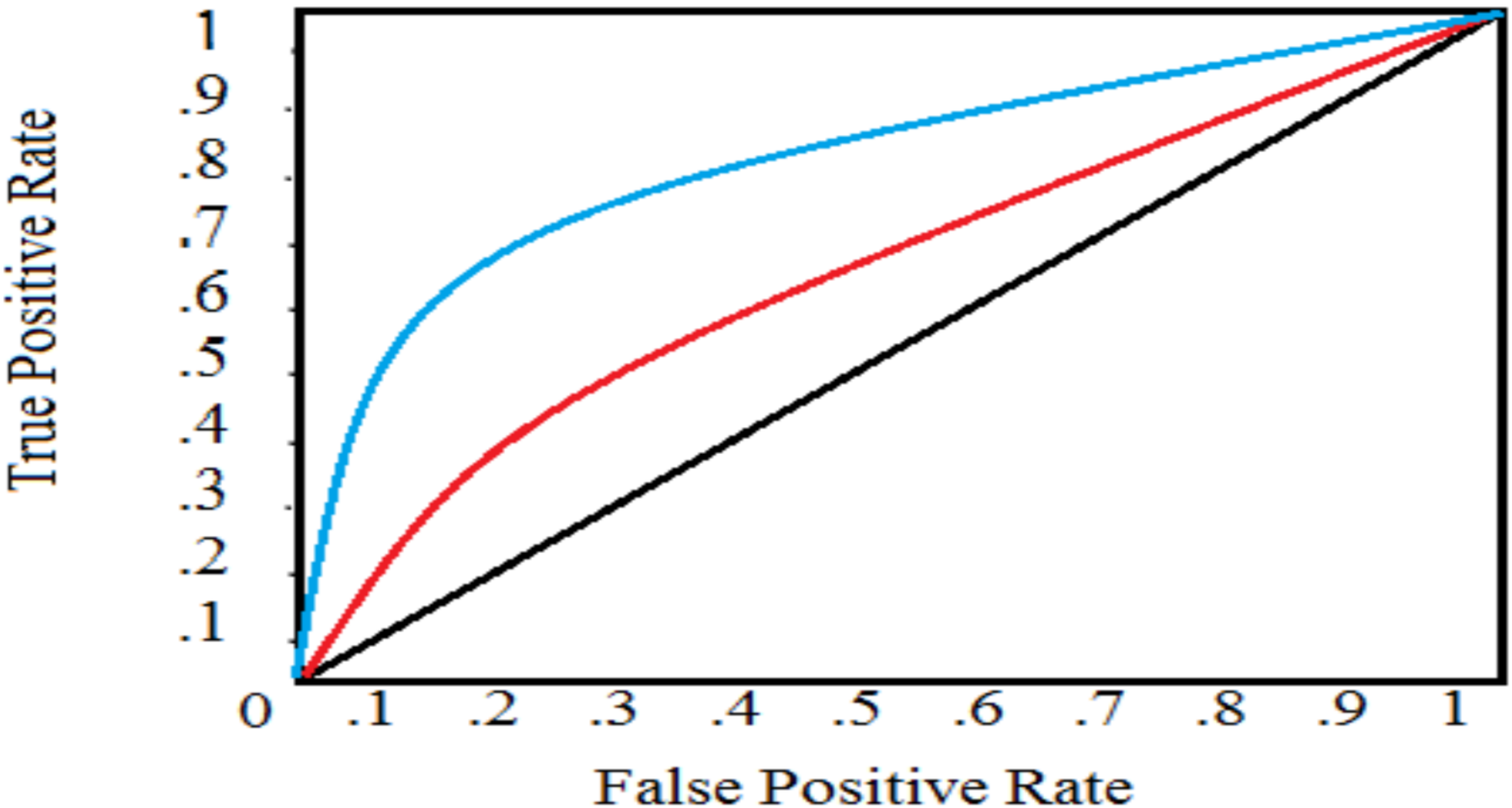
$$\text{false positive rate} = \frac{\mathrm{FP}}{\text{actual neg}} = \frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}$$

ROC/AUC

- A Receiver Operating Characteristic (ROC)/Area Under Curve plots the TP-rate vs. the FP-rate as a threshold on the confidence of an instance being positive is varied.

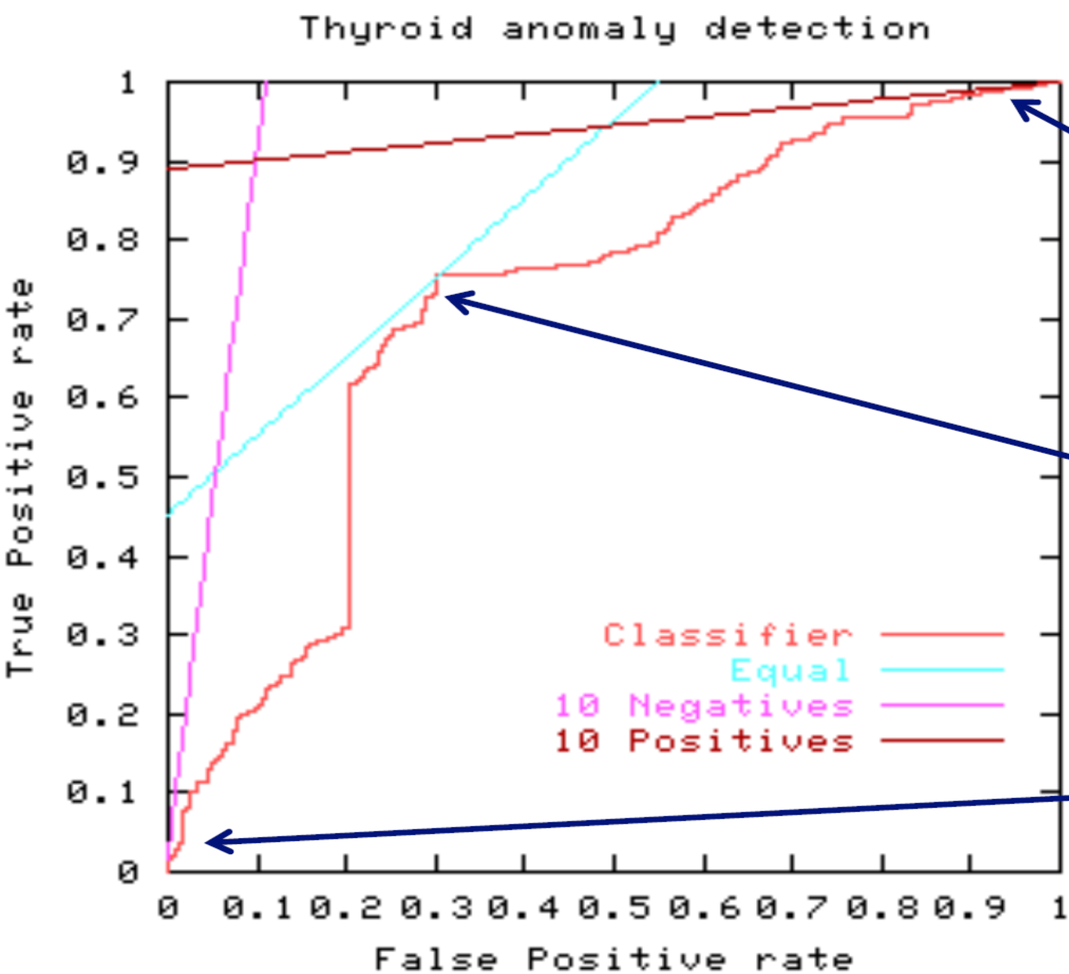


Example Curve of ROC/AUC



The principal advantage of the AUC is that it is more robust than Accuracy in class imbalanced situations

ROC curves & Misclassification costs



Best operating point when ***FN*** costs 10× ***FP***

Best operating point when cost of misclassifying positives and negatives is equal

Best operating point when FP costs 10× FN

Step to create ROC

- Sort test-set predictions according to confidence that each instance is positive.

- Step through sorted list from high to low confidence

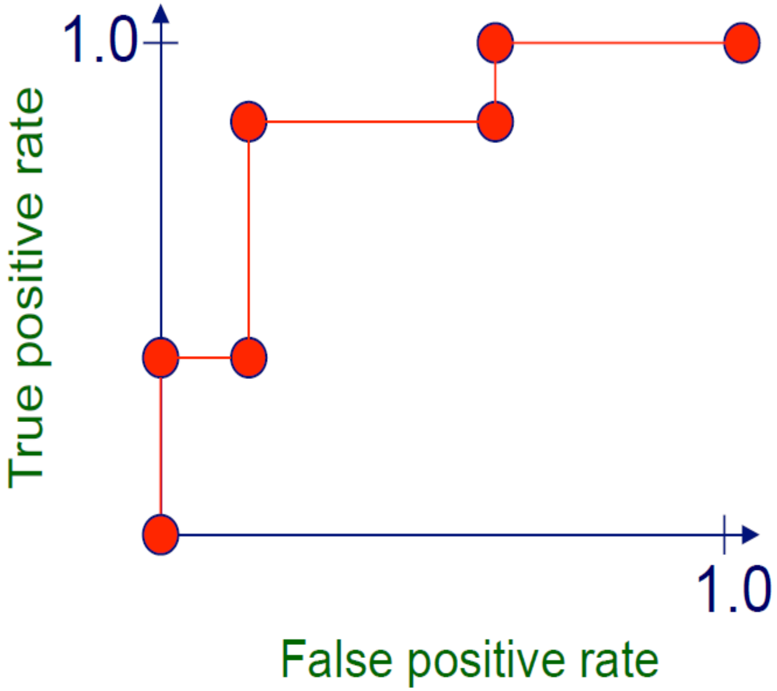
  - ✓ locate a *threshold* between instances with opposite classes (keeping instances with the same confidence value on the same side of threshold)

  - ✓ compute TPR, FPR for instances above threshold

  - ✓ output (FPR, TPR) coordinate

Example of ROC Plot

instance	confidence positive		correct class
Ex 9	.99		+
Ex 7	.98	TPR= 2/5, FPR= 0/5	+
Ex 1	.72	TPR= 2/5, FPR= 1/5	-
Ex 2	.70		+
Ex 6	.65	TPR= 4/5, FPR= 1/5	+
Ex 10	.51		-
Ex 3	.39	TPR= 4/5, FPR= 3/5	-
Ex 5	.24	TPR= 5/5, FPR= 3/5	+
Ex 4	.11		-
Ex 8	.01	TPR= 5/5, FPR= 5/5	-



Example of ROC Plot …

- **Rearrange the samples according to class**

Correct class	Instance	Confidence Positive
+	Ex 9	0.99
+	Ex 7	0.98
+	Ex 2	0.70
+	Ex 6	0.65
+	Ex 5	0.24
-	Ex 1	0.72
-	Ex10	0.51
-	Ex 3	0.39
-	Ex 4	0.11
-	Ex 8	0.01

Positive Class

Negative Class

Example of ROC Plot …

- **For Threshold 0.72**

Correct class	Instance	confidence positive	predicted class
+	Ex 9	0.99	+
+	Ex 7	0.98	+
+	Ex 2	0.70	-
+	Ex 6	0.65	-
+	Ex 5	0.24	-
-	Ex 1	0.72	+
-	Ex10	0.51	-
-	Ex 3	0.39	-
-	Ex 4	0.11	-
-	Ex 8	0.01	-

Confidence > threshold
Positive class
Else
Negative class

TP=2
FP=1
TN=4
FN=3
TPR=TP/TP+FN=2/5
FPR=FP/FP+TN=1/5

Example of ROC Plot …

- **For Threshold 0.65**

Correct class	Instance	confidence positive	predicted class
+	Ex 9	0.99	+
+	Ex 7	0.98	+
+	Ex 2	0.70	+
+	Ex 6	0.65	+
+	Ex 5	0.24	-
-	Ex 1	0.72	+
-	Ex10	0.51	-
-	Ex 3	0.39	-
-	Ex 4	0.11	-
-	Ex 8	0.01	-

Confidence > threshold
Positive class
Else
Negative class

TP=4
FP=1
TN=4
FN=1
TPR=TP/TP+FN=4/5
FPR=FP/FP+TN=1/5

ROC Plot…

- Can interpolate between points to get *convex hull*
  - ✓ Convex hull: repeatedly, while possible, perform interpolations that skip one data point and discard any point that lies below a line
  - ✓ Interpolated points are achievable in theory: can flip weighted coin to choose between classifiers represented by plotted points

ROC Curve

- Does a low false-positive rate indicate that most positive predictions (i.e. predictions with confidence > some threshold) are correct?
- Consider: TPR is 0.9, and FPR is 0.01

Fraction of instances that are positive	Fraction of positive predictions that are correct
0.5	0.989
0.1	0.909
0.01	0.476
0.001	0.083

Issues with ROC/AUC

- AUC/ROC has adopted as replacement of accuracy but it has also some criticism such as:
  - The ROC curves on which the AUCs of different classifiers are based may cross, thus not giving an accurate picture of what is really happening.
  - The misclassification cost distributions used by the AUC are different for different classifiers.
  - Therefore, we may be comparing “apples and oranges” as the AUC may give more weight to misclassifying a point by classifier A than it does by classifier B. Ans: H-Measure

Other Accuracy Metrics

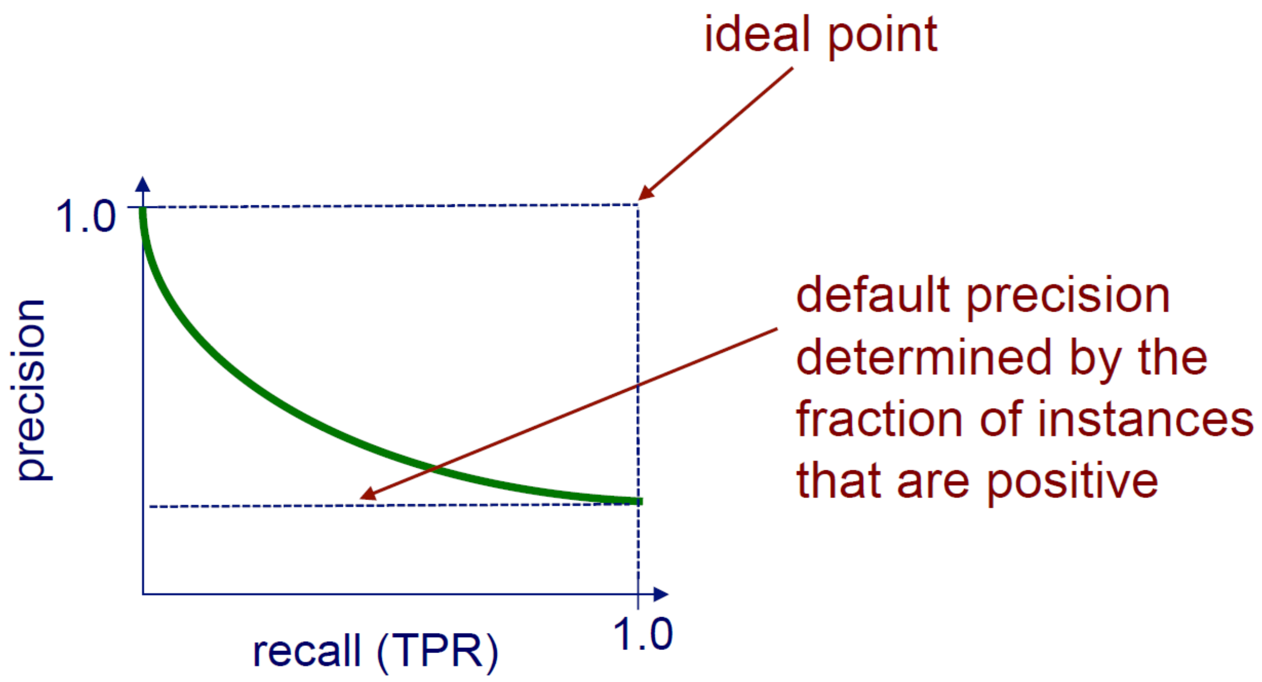
		actual class	
		positive	negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

$$\text{recall (TP rate)} = \frac{\text{TP}}{\text{actual pos}} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{precision} = \frac{\text{TP}}{\text{predicted pos}} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Precision/recall curves

- A *precision/recall curve* plots the precision vs. recall (TP-rate) as a threshold on the confidence of an instance being positive is varied.



Comment on ROC/PR Curve

- Both
  - allow predictive performance to be assessed at various levels of confidence
  - assume binary classification tasks
  - sometimes summarized by calculating *area under the curve*
- ROC curves
  - insensitive to changes in class distribution (ROC curve does not change if the proportion of positive and negative instances in the test set are varied)
  - can identify optimal classification thresholds for tasks with differential misclassification costs
- Precision/Recall curves
  - show the fraction of predictions that are false positives
  - well suited for tasks with lots of negative instances