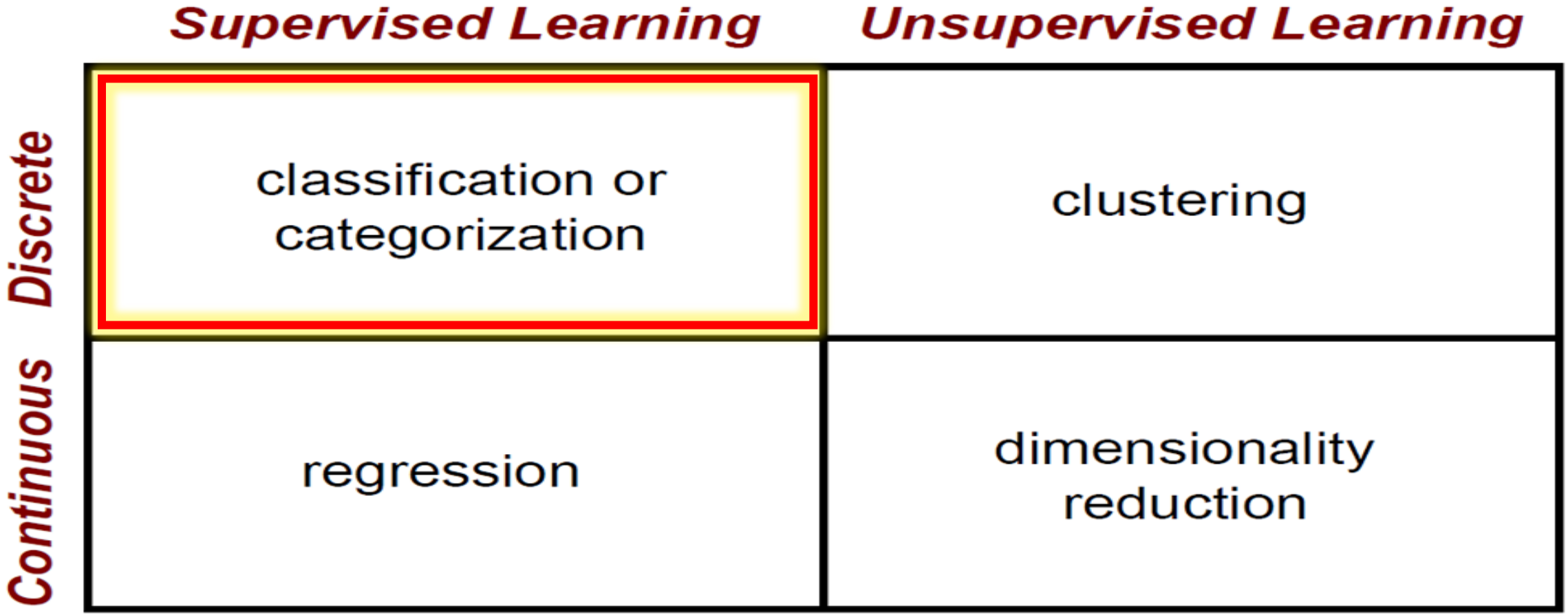
Introduction to Machine Learning

Classifier Evaluation

ML Problems: Recall



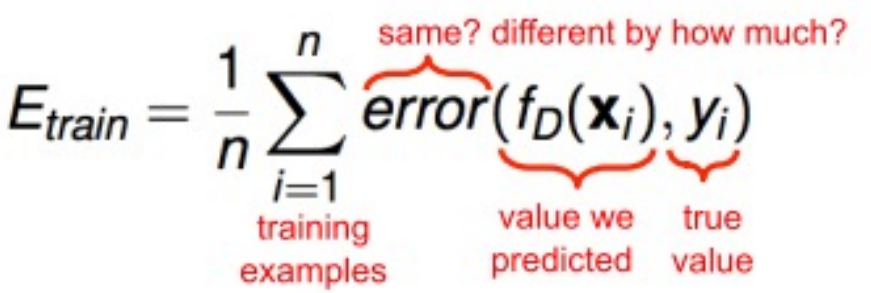
Classification Methods

- k-Nearest Neighbors
- Decision Trees
- Naïve Bayes
- Support Vector Machines
- Logistic Regression
- Neural Networks
- Ensemble Methods (Boosting, Random Forests)

How to evaluate?

Training vs Generalization Error

- Training Error
  - Not very useful
  - Relatively easy to obtain low error



Empirical Risk Minimization

- Generalization Error
  - How well we do on future data

How to compute generalization error?

Estimating Generalization Error

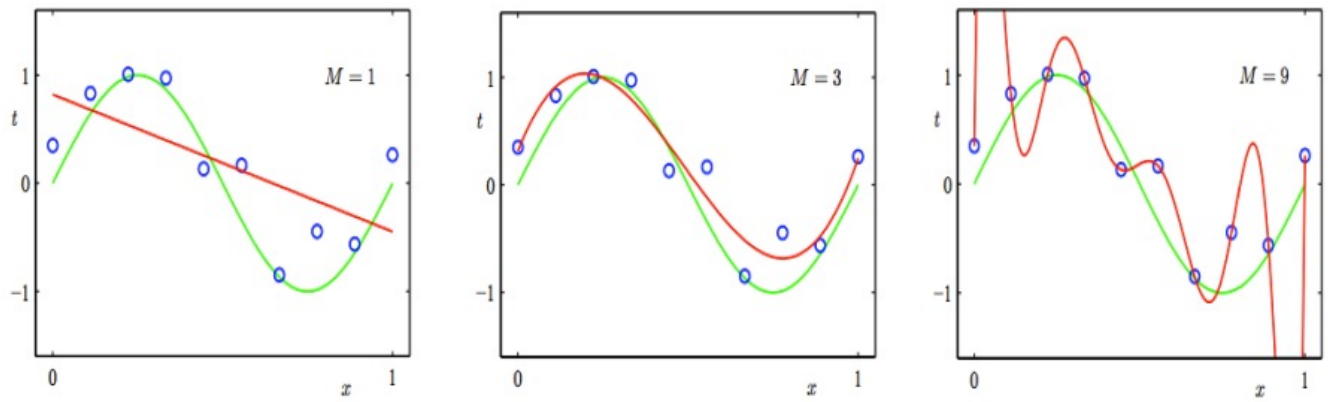
- Testing Error
  - Set aside part of training data (testing set)
  - Learn a predictor without using any of this test data
  - Predict values for testing set, compute error
  - This is an estimate of generalization error

$$E_{test} = \frac{1}{n} \sum_{i=1}^{n} error(f_D(\mathbf{x}_i), y_i)$$

over testing set

Underfitting and Overfitting

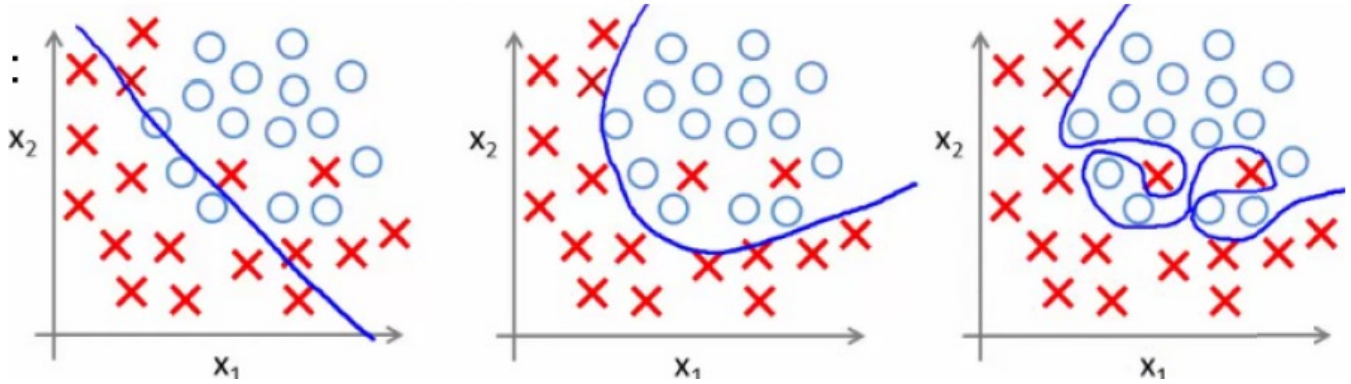
Regression



predictor too inflexible:
cannot capture pattern

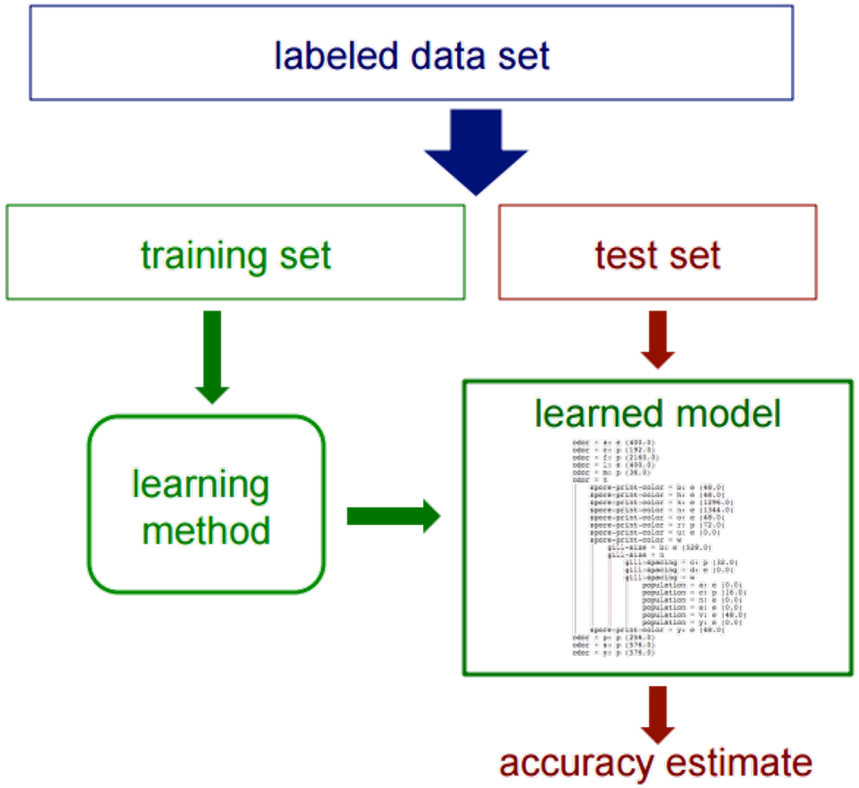
predictor too flexible:
fits noise in the data

Classification

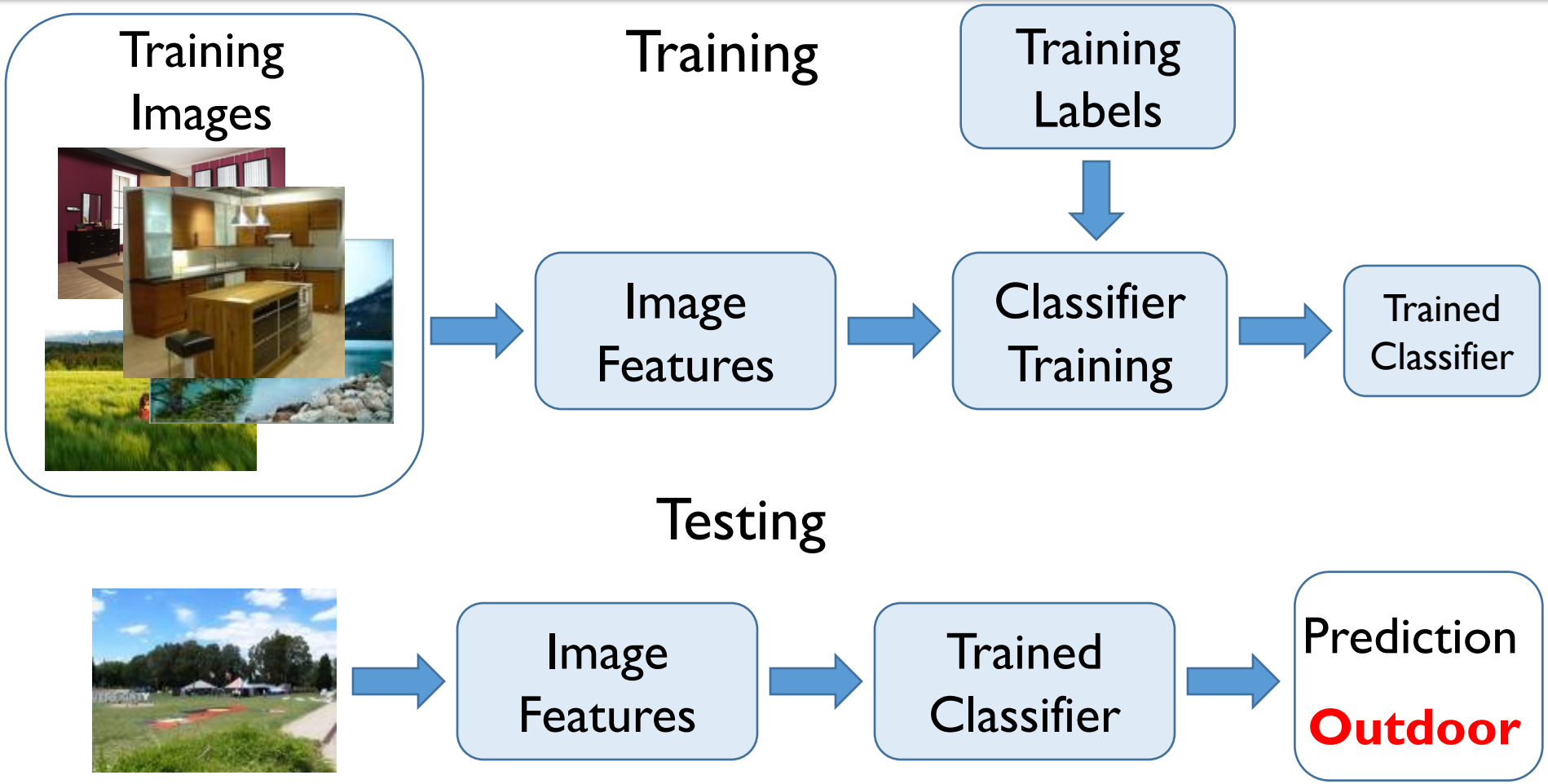


Estimating Generalization Error

- Getting an unbiased estimate of the accuracy of a learned model



Example: Image Classification



Source: Derek Hoiem

Training, Validation, Test Sets

Training set

- NB: Count frequencies, DT: Pick attributes to split on

Validation set

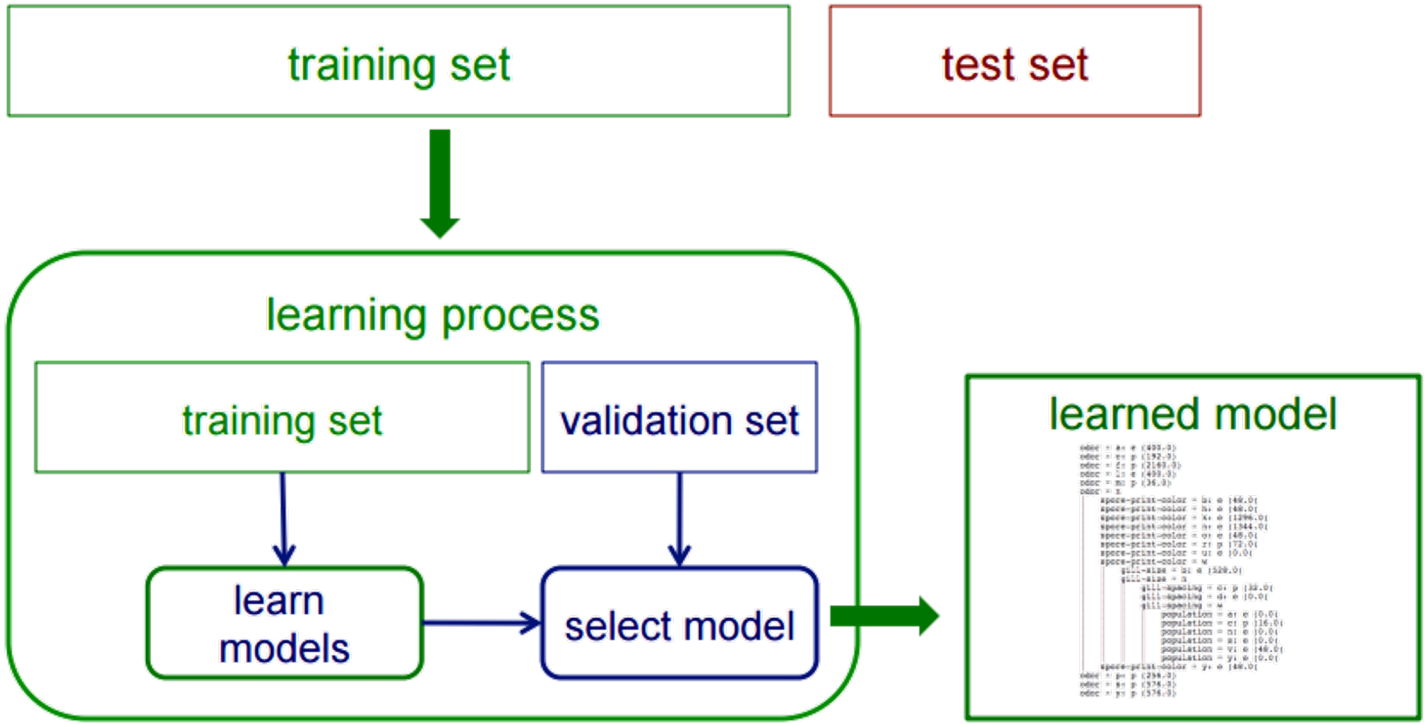
- Pick best-performing algorithm (NB vs DT vs..)
- Fine-tune parameters (Tree depth, k in kNN, c in SVM)

Testing set

- Run multiple trials and average

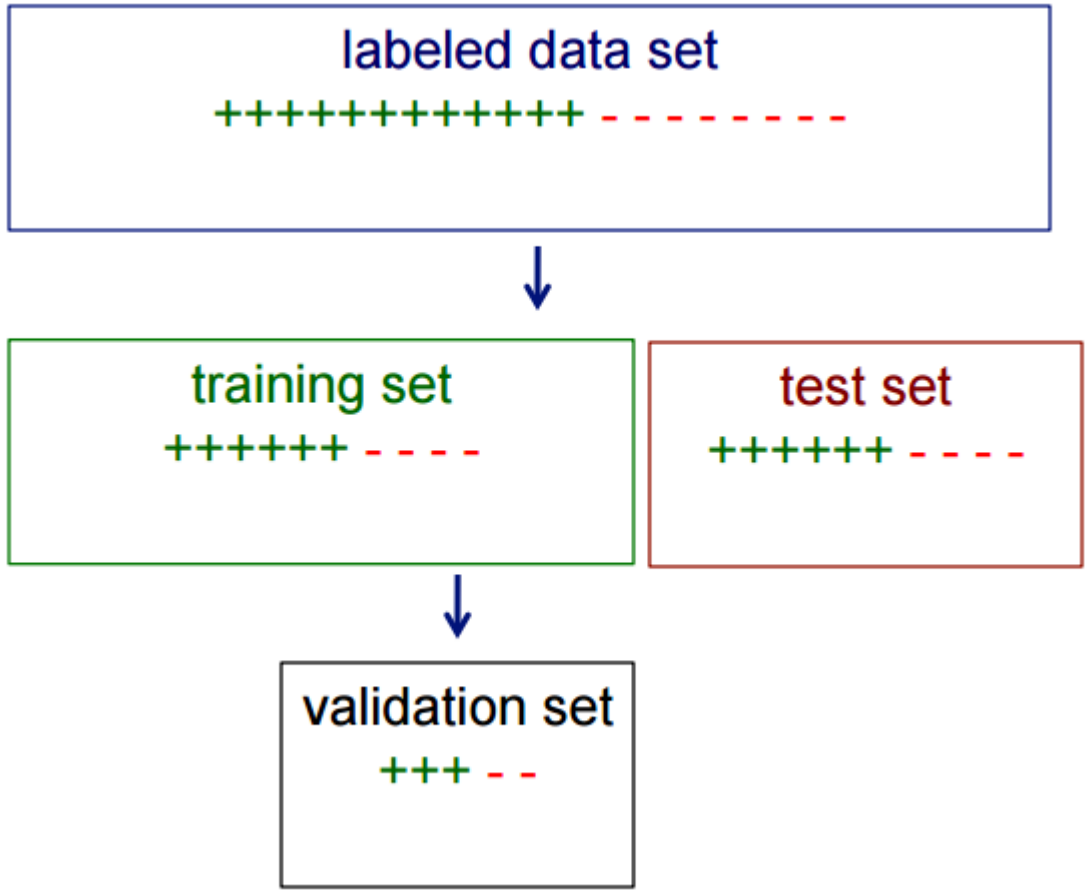
Use of Validation Sets

- If we want unbiased estimates of accuracy during the learning process:



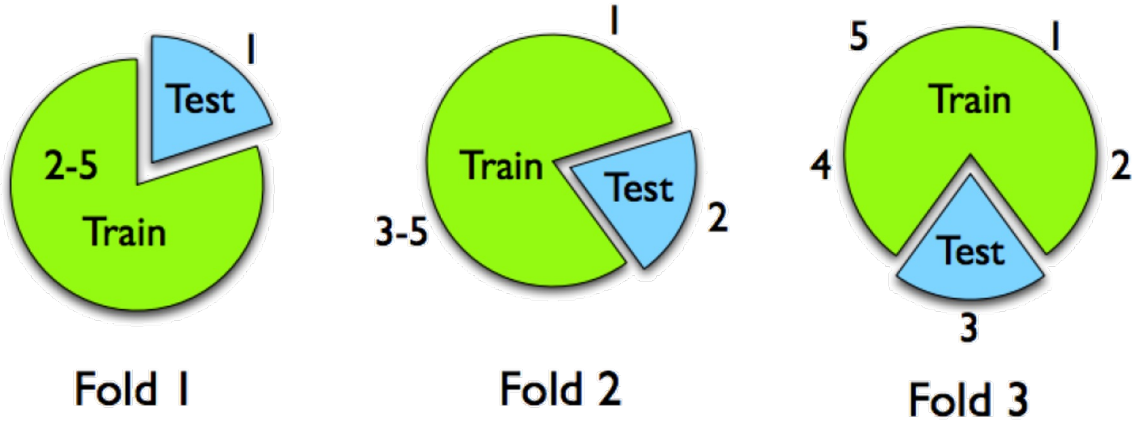
Stratified Sampling

- 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.



Model Selection

- Resubstitution
- K-fold cross-validation



- Leave-one-out
  - N-fold cross-validation

Cross-Validation: Example

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

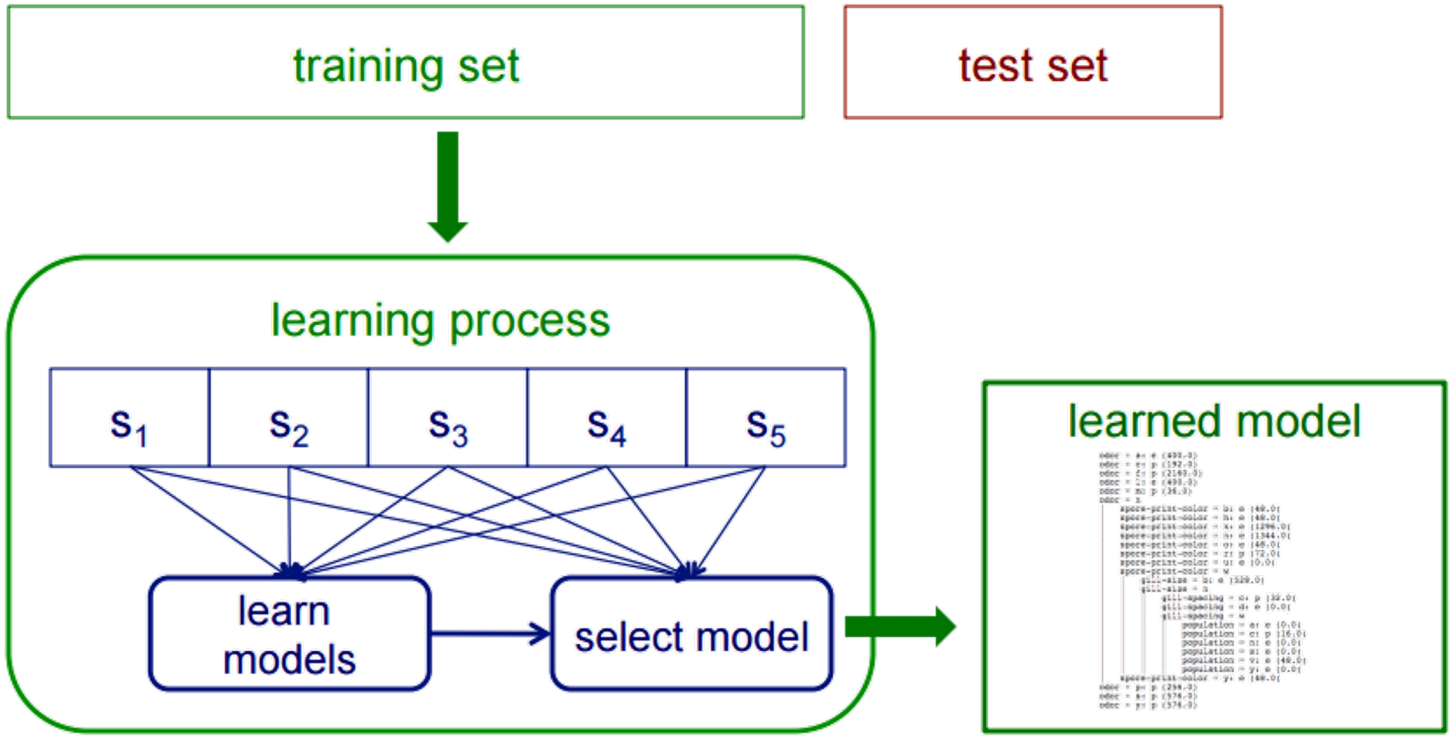
iteration	train on	test on	correct
1	S_2 S_3 S_4 S_5	S_1	11 / 20
2	S_1 S_3 S_4 S_5	S_2	17 / 20
3	S_1 S_2 S_4 S_5	S_3	16 / 20
4	S_1 S_2 S_3 S_5	S_4	13 / 20
5	S_1 S_2 S_3 S_4	S_5	16 / 20

Classification Accuracy = 73/100 = 73%

Note: 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

Cross-Validation: Example

- 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 k in k-NN)



Evaluation Measures

- Classification
  - How often we classify something right/wrong
- Regression
  - How close are we to what we're trying to predict
- Ranking/Search
  - How correct are the top-k results?
- Clustering
  - How well we describe our data (Not straightforward)

Regression Error: Root Mean Square Error

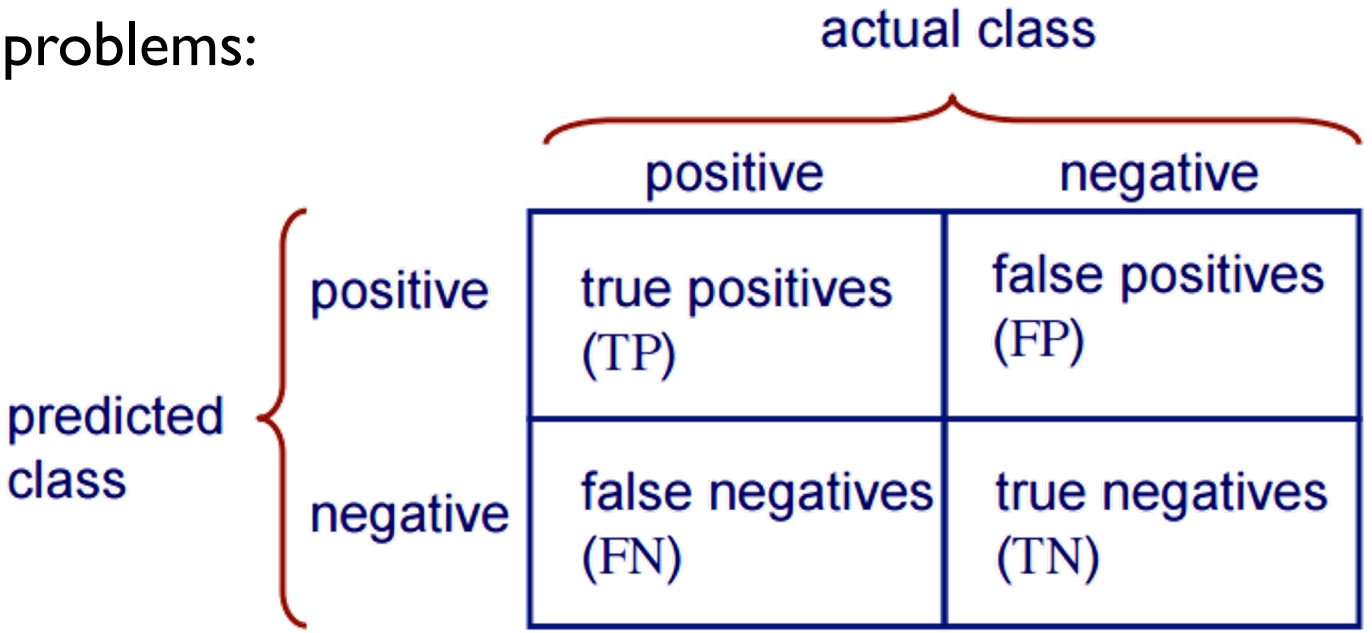
Regression consider real valued output

Regression

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Classification Error: Accuracy

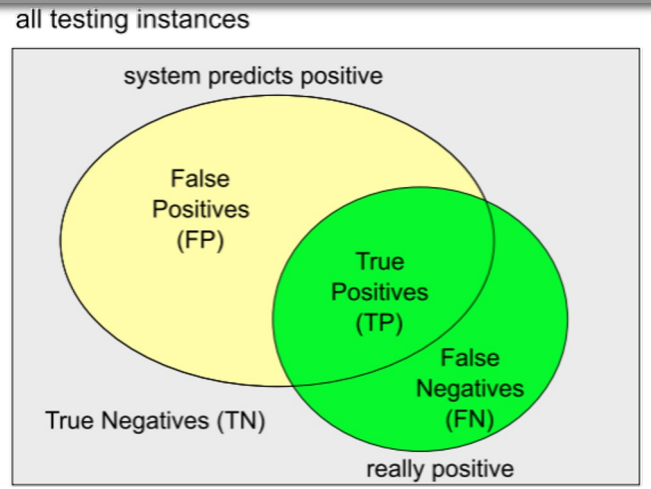
In 2-class problems:



$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Classification Performance Measures

Really positive? \ Predict positive?	Yes	No
Yes	TP	FN
No	FP	TN



- True Positive Rate also called "Sensitivity"
- "Specificity" = 1 – False Alarm

- Classification Error: $\frac{errors}{total} = \frac{FP+FN}{TP+TN+FP+FN}$
- Accuracy = 1-Error: $\frac{correct}{total} = \frac{TP+TN}{TP+TN+FP+FN}$

meaningless if classes imbalanced

- False Alarm = False Positive rate = FP / (FP+TN)
- Miss = False Negative rate = FN / (TP+FN)
- Recall = True Positive rate = TP / (TP+FN)
- Precision = TP / (TP+FP)

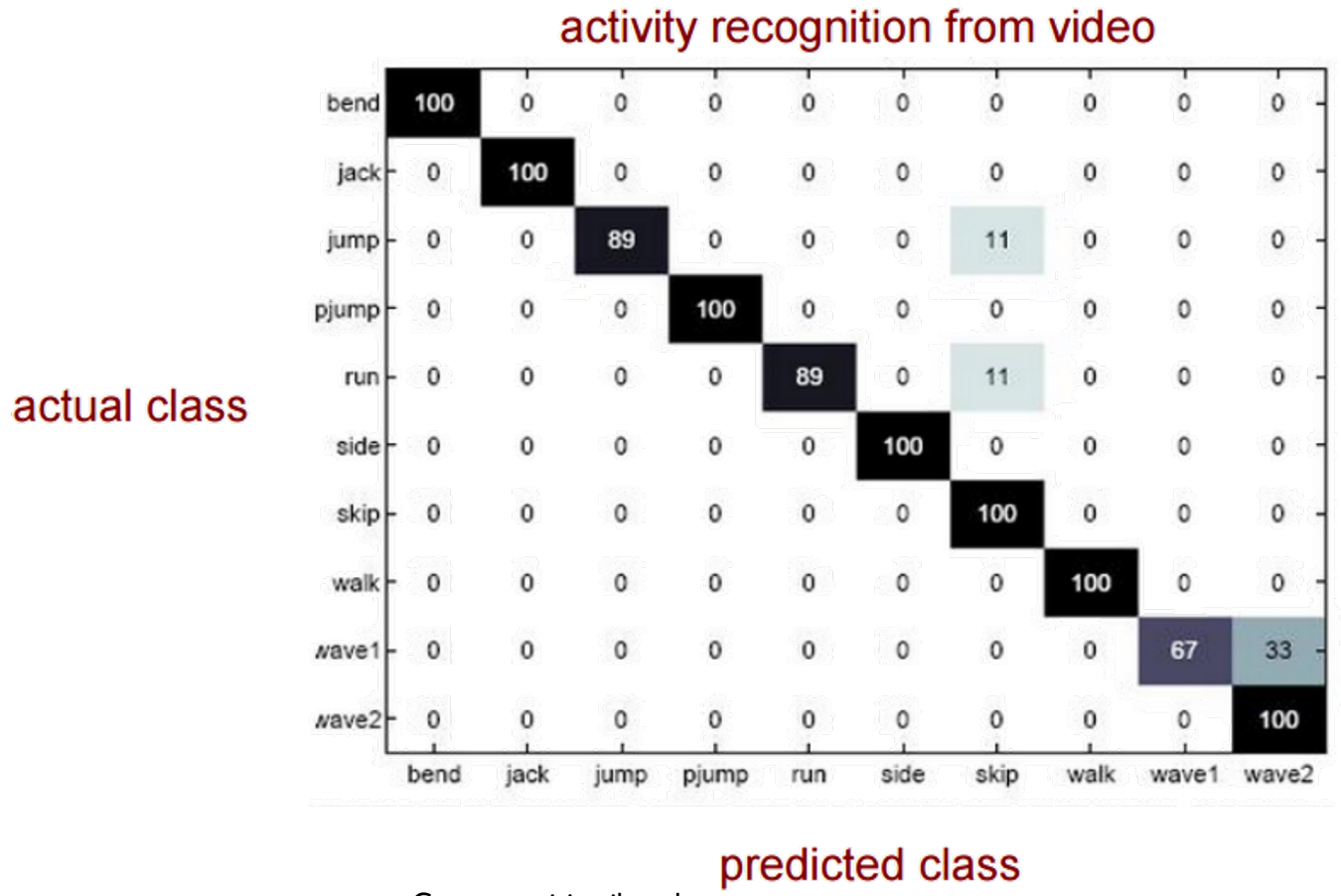
always report in pairs, e.g.: Miss / FA or Recall / Prec.

- "Sensitivity" = Probability of a positive test given a patient has the disease
- "Specificity" = Probability of a negative test given a patient is well

Classification Error: Beyond Accuracy

For multi-class problems?

Confusion Matrix



Courtesy: vision.jhu.edu

Is accuracy adequate?

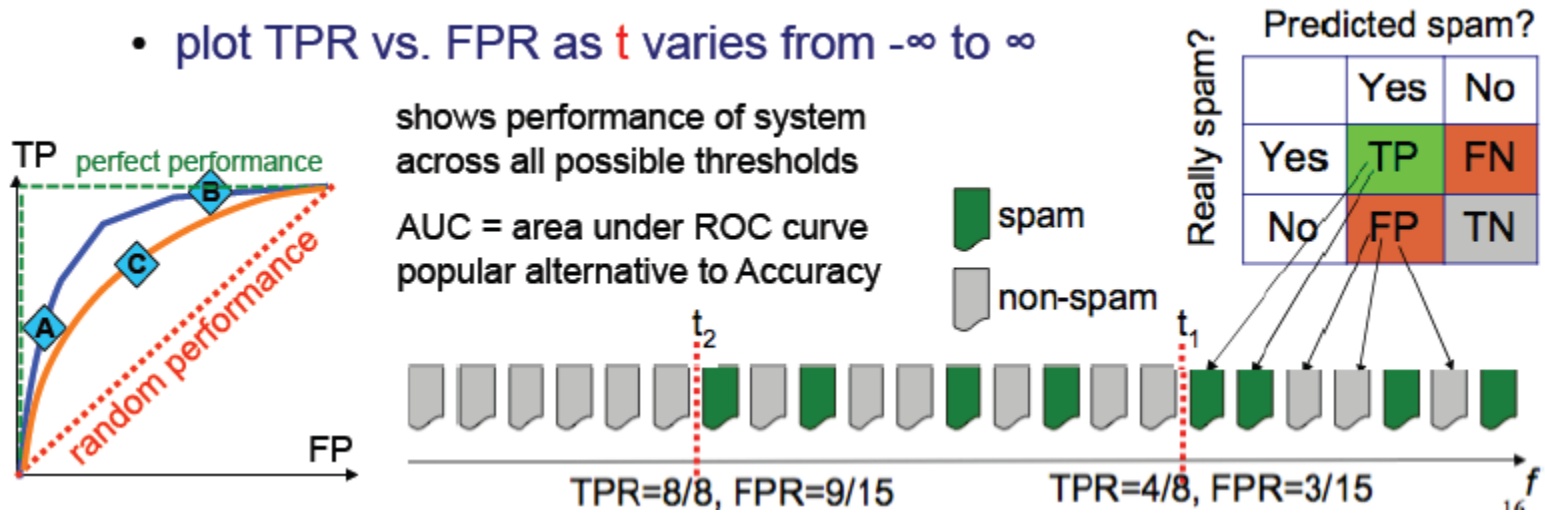
- Accuracy may not be useful in cases where
  - There is a large class skew
    - Is 98% accuracy good if 97% of the instances are negative?
  - There are differential misclassification costs – say, getting a positive wrong costs more than getting a negative wrong
    - Consider a medical domain in which 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

Utility and Cost

- Sometimes, there is a cost for each error
  - E.g. Earthquake prediction
    - False positive: Cost of preventive measures
    - False negative: Cost of recovery
- Detection Cost (Event detection)
  - Cost = C_{FP} * FP + C_{FN} * FN
- F-measure (Information Retrieval)
  - F1 = 2/(1/Recall + 1/Precision)

ROC Curves

- Many algorithms compute "confidence" f(x)
  - Threshold to get decision: spam if f(x) > t, non-spam if f(x) <= t
  - Threshold to determine error rates
- Receiver Operating Characteristic (ROC)

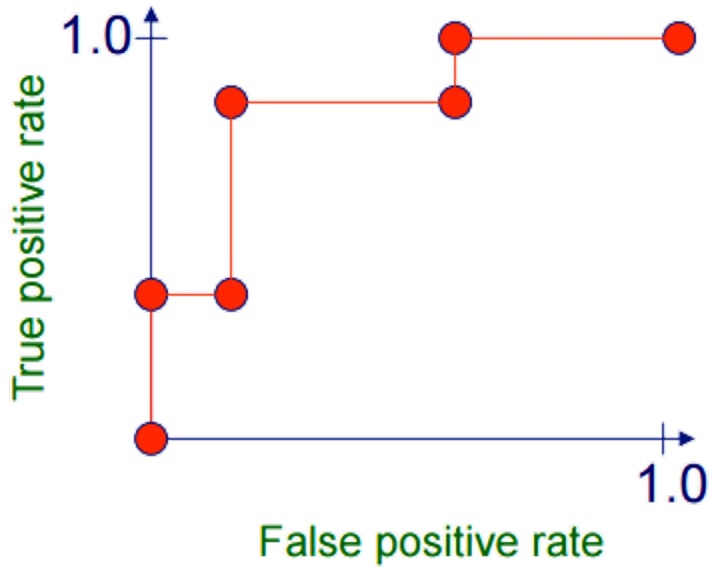


ROC Curve: Algorithm

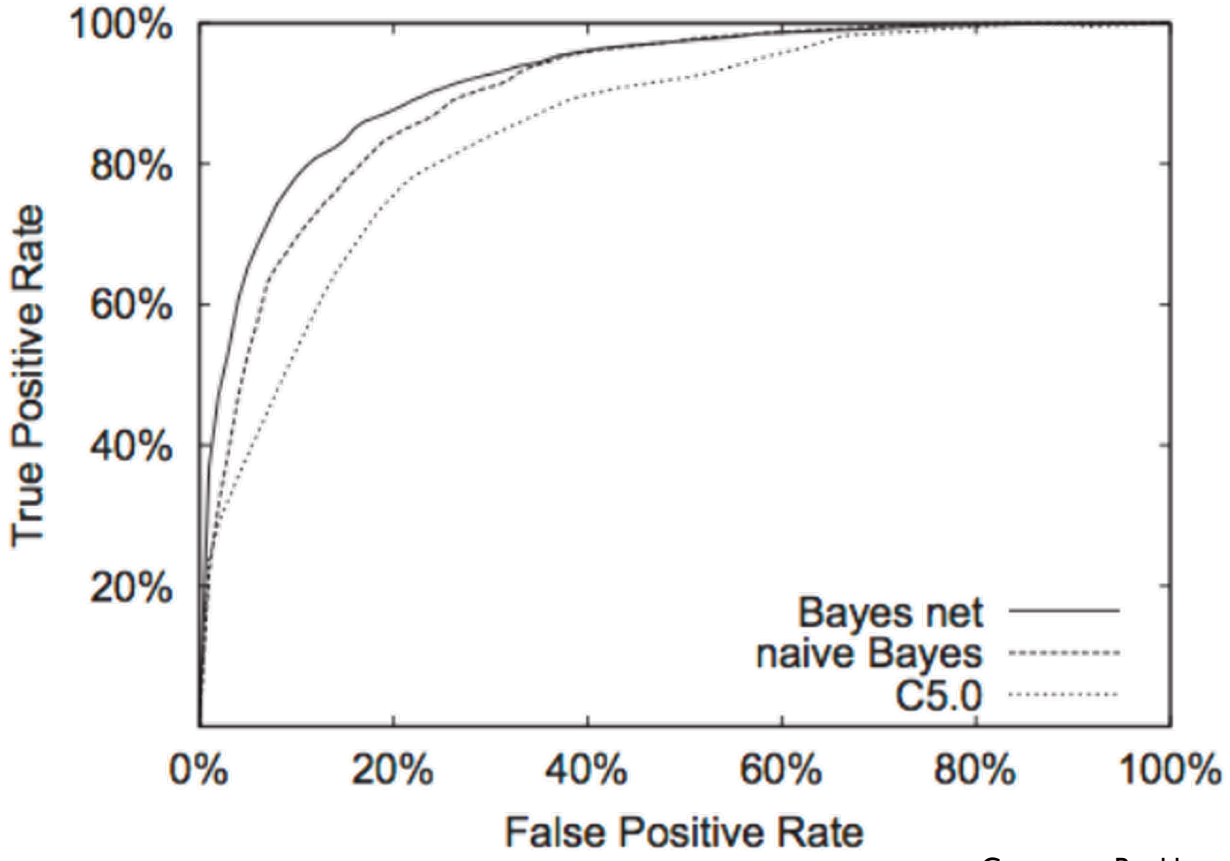
- Sort test-set predictions according to confidence that each instance is positive
- Step through sorted list from high to low confidence
  - Locate a threshold between instances with opposite classes (keeping instances with the same confidence value on the same side of threshold)
  - Compute TPR, FPR for instances above threshold
  - Output (FPR, TPR) coordinate

Plotting an ROC Curve

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	-

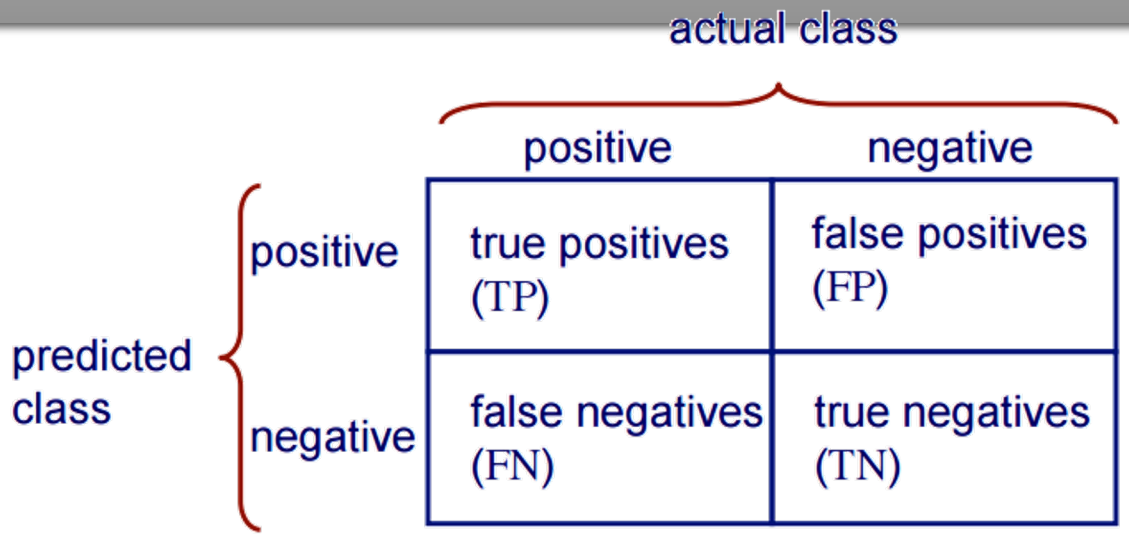


ROC Curve: Example



Courtesy: Bockhorst et al., Bioinformatics 2003

Recall: Precision-Recall



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

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

F-score: Harmonic mean of precision and recall
(2 * precision * recall)/(precision+recall)

Summary

- Rigorous statistical evaluation is extremely important in experimental computer science in general and machine learning in particular
- How good is a learned hypothesis?
- How close is the estimated performance to the true performance?
- Is one hypothesis better than another?
- Is one learning algorithm better than another on a particular learning task?

References

- Key References
  - Chapter 19, EA Introduction to ML, 2nd Edn
  - Chapter 1 (Sec 1.1-1.5), Pattern Recognition and Machine Learning, Bishop
- Other Recommended References
  - http://www.icmla-conference.org/icmla11/PE_Tutorial.pdf (Tutorial on Performance Evaluation of Classifiers)
  - Chapter 5 ('Evaluating Hypotheses'), Machine Learning by Tom Mitchell
    - http://www.cs.cmu.edu/~tom/mlbook.html