ML410C

Projects in health informatics – Project and information management

Data Mining

Last time…

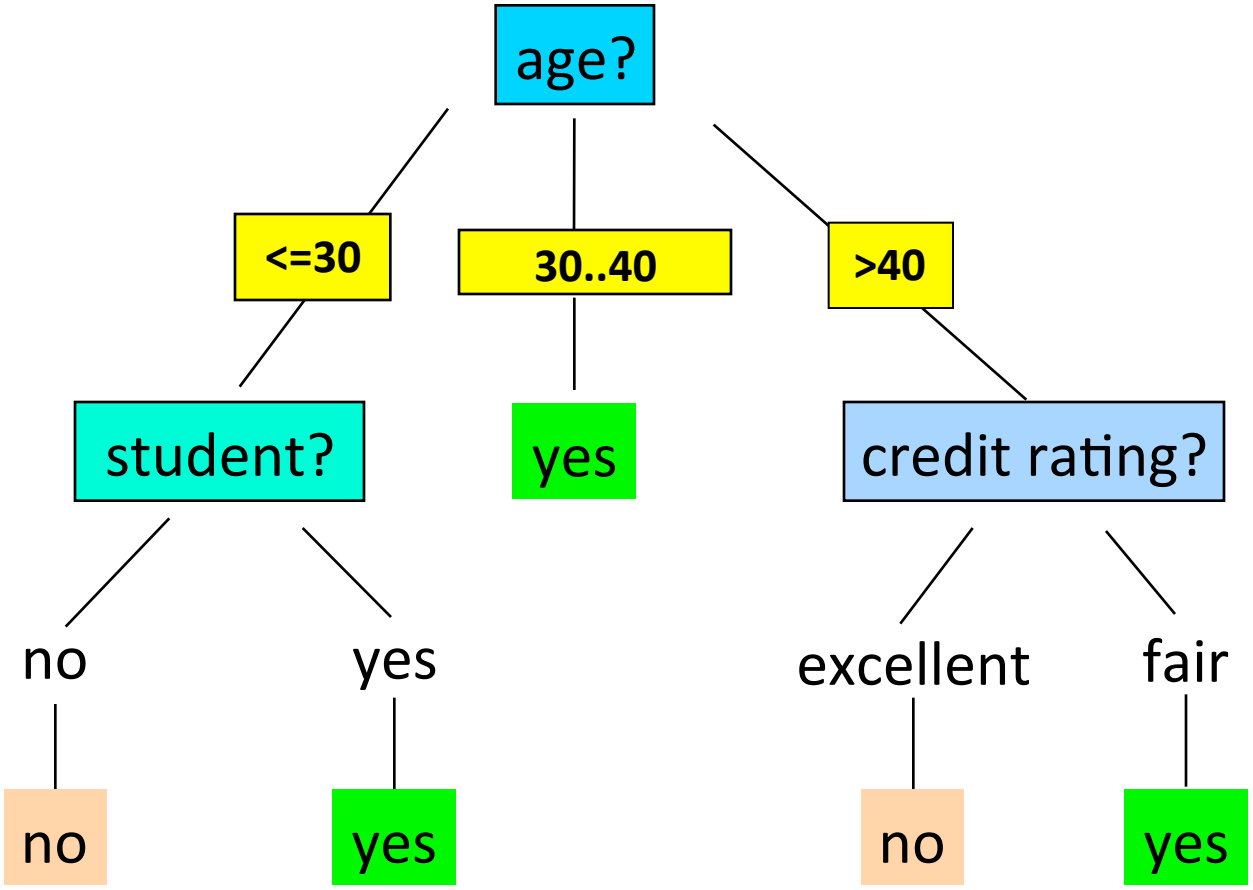
- What is classification
- Overview of classification methods
- Decision trees
- Forests

Training Dataset

Example

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Output: A Decision Tree for "*buys_computer*"



Today

DATE	TIME	ROOM	TOPIC
MONDAY 2013-09-09	10:00-11:45	502	Introduction to data mining
WEDNESDAY 2013-09-11	09:00-10:45	501	Decision trees, rules and forests
FRIDAY 2013-09-13	10:00-11:45	Sal C	Evaluating predictive models and tools for data mining

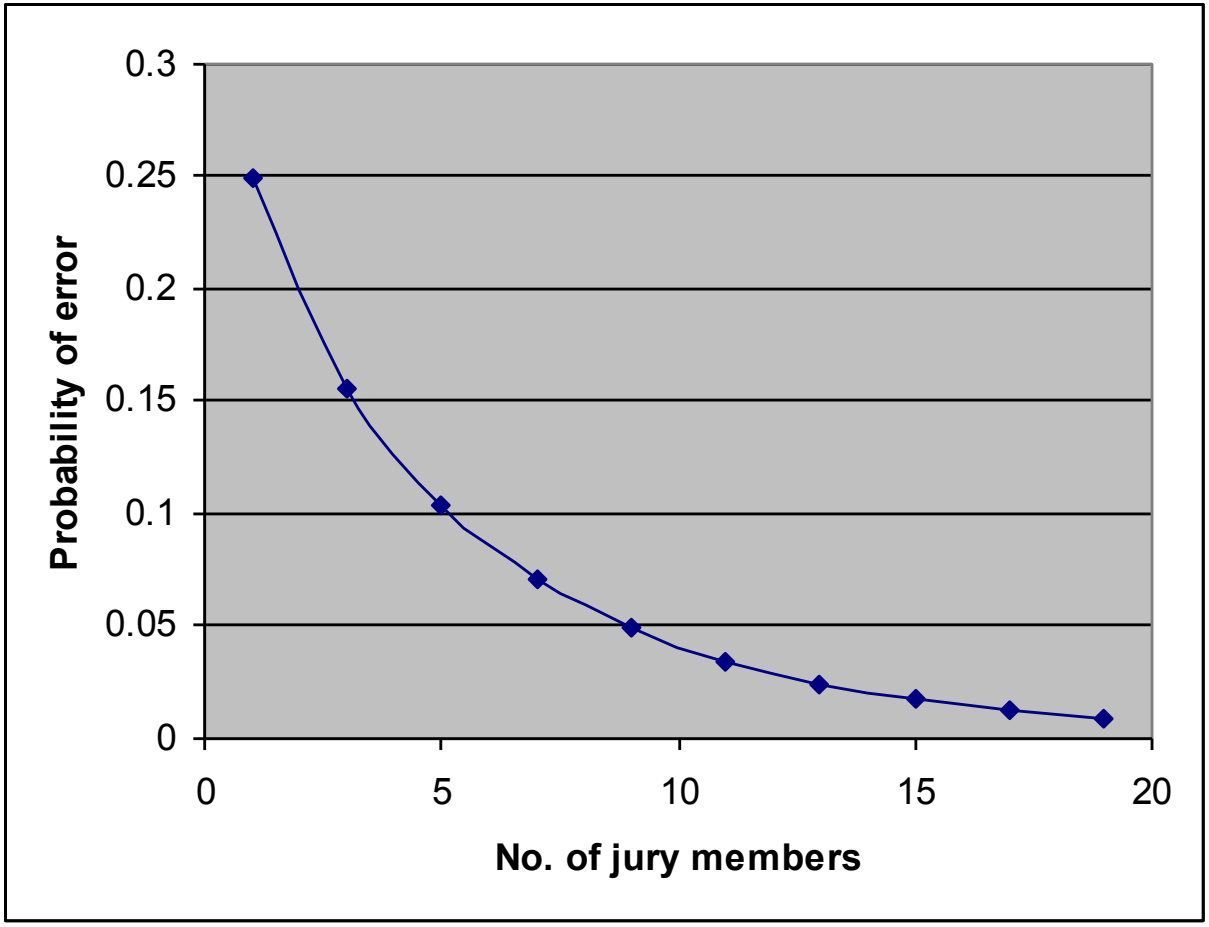
Today

- Forests (finishing up)
- Accuracy
- Evaluation methods
- ROC curves

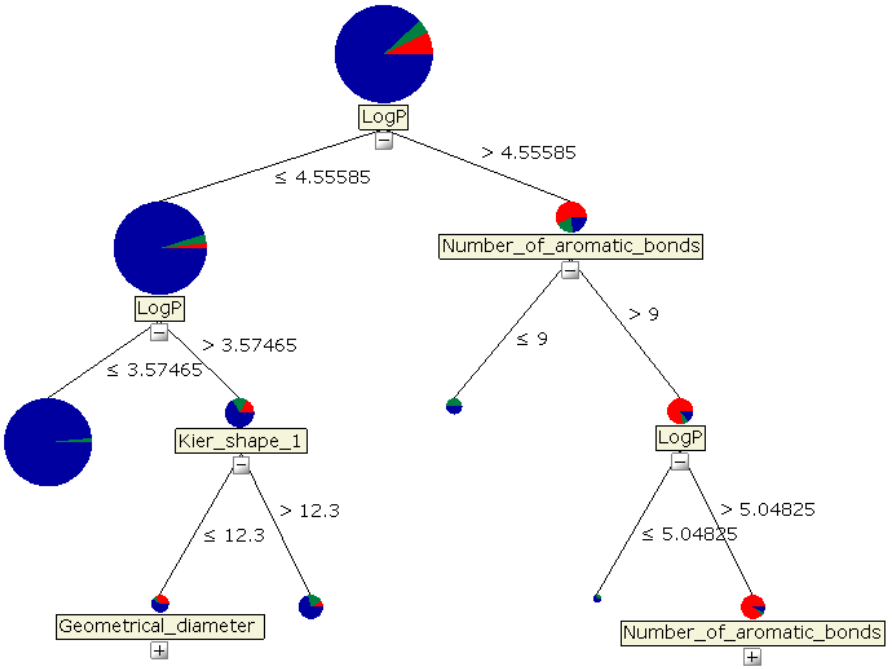
Condorcet's jury theorem

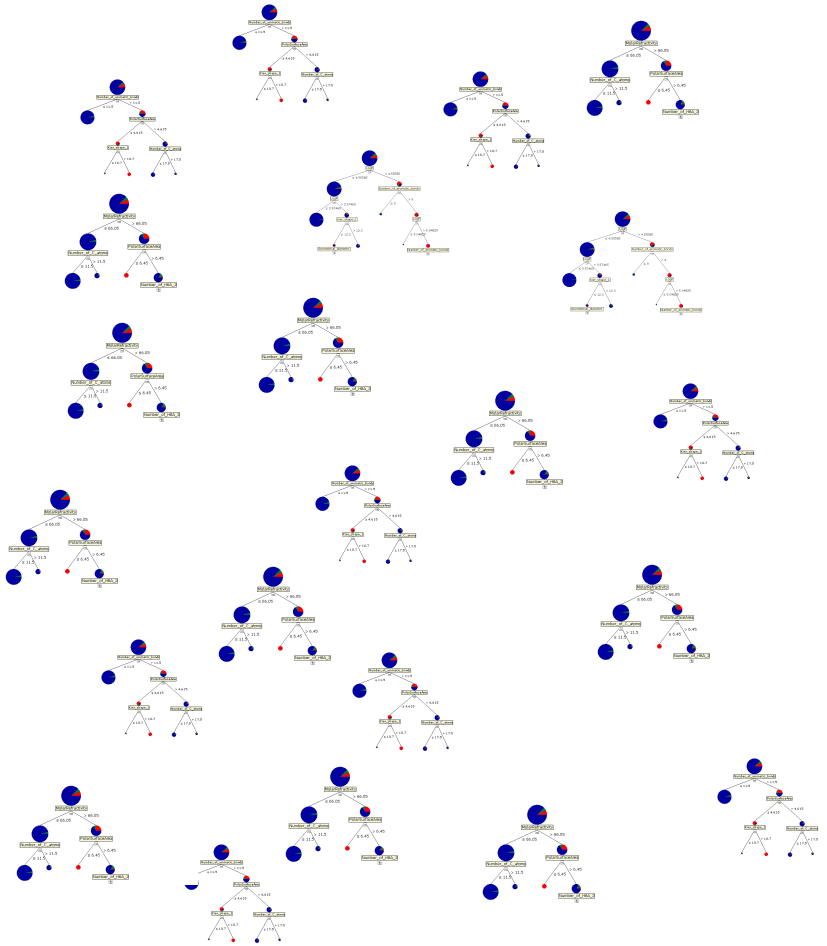
- *If each member of a jury is more likely to be right than wrong,*
- *then the majority of the jury, too, is more likely to be right than wrong*
- *and the probability that the right outcome is supported by a majority of the jury is a (swiftly) increasing function of the size of the jury,*
- *converging to 1 as the size of the jury tends to infinity*

Condorcet's jury theorem



Single trees vs. forests





Bagging

- Also known as bootstrapping…
- E: a set of samples, with $n = |E|$
- A *bootstrap replicate E'* of *E* is created by randomly selecting *n* examples from *E* with replacement

The probability of an example in *E* appearing in *E'* is

$$1 - \left(\frac{n-1}{n}\right)^n = 1 - \frac{1}{e} \approx 0.632$$

Breiman L., "Bagging Predictors", *Machine Learning*, vol. 24 (1996) 123-140

Bootstrap replicate

Ex.	Other	Bar	Fri/Sat	Hungry	Guests	Wait
e2	yes	no	no	yes	full	no
e2	yes	no	no	yes	full	no
e3	no	yes	no	no	some	yes
e4	yes	no	yes	yes	full	yes
e4	yes	no	yes	yes	full	yes
e6	no	yes	no	yes	some	yes

Bagging

Input: examples E, base learner BL, iterations n

Output: combined model M

i:= 0

Repeat

i := i+1

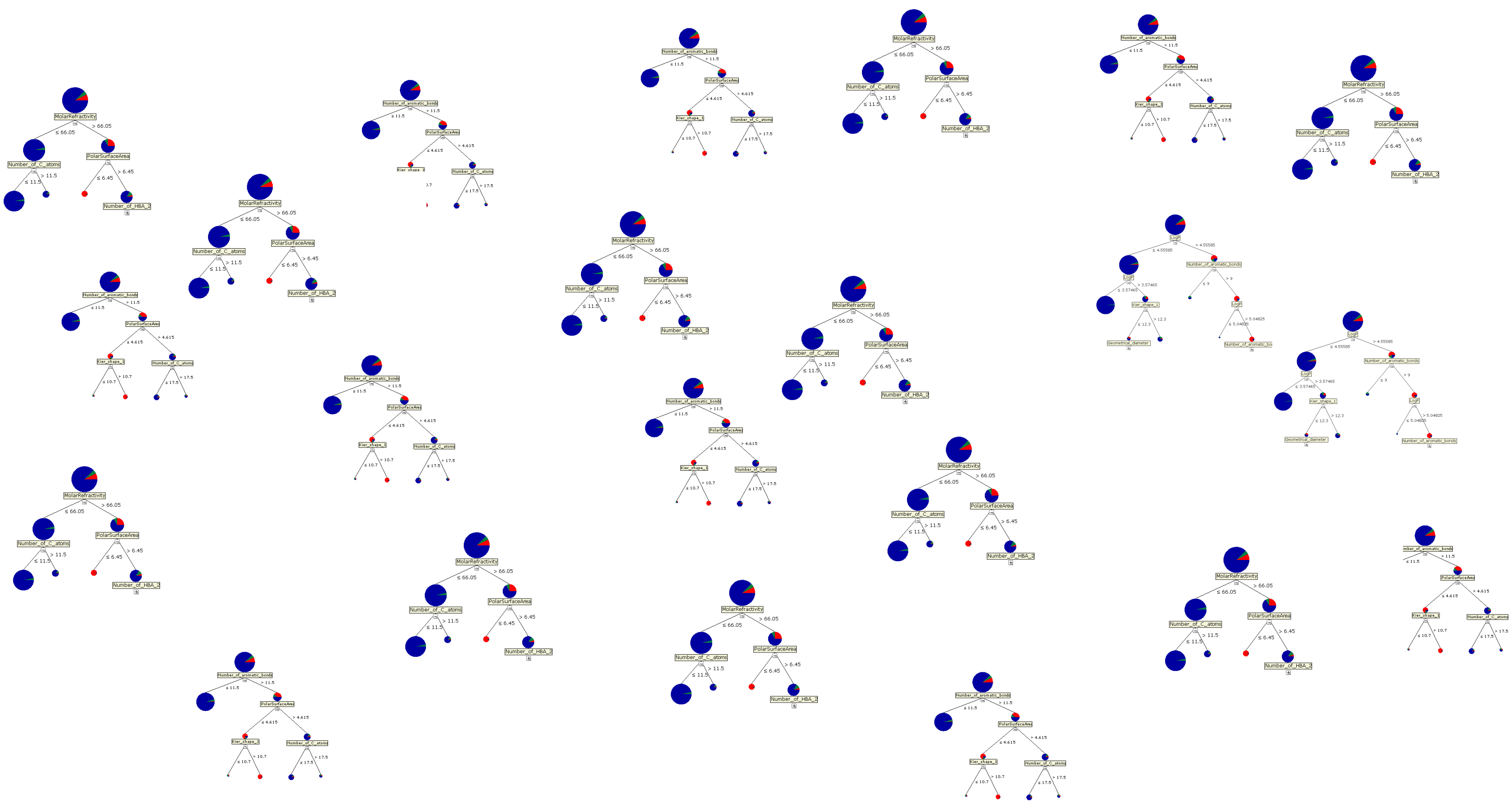
Generate *bootstrap replicate* E' of E

M_i := $BL(E')$

Until i = N

M := majority vote($\{M_1, \ldots, M_n\}$)

Forests



Model Evaluation

- Metrics for Performance Evaluation
  - How to evaluate the performance of a model?
- Methods for Performance Evaluation
  - How to obtain reliable estimates?
- Methods for Model Comparison
  - How to compare the relative performance of different models?

Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - Rather than how fast it takes to classify or build models, scalability, etc.
- Confusion Matrix:

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a: TP	b: FN
	Class=No	c: FP	d: TN

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Accuracy: ?

Metrics for Performance Evaluation

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Limitation of Accuracy

- Consider a 2-class problem
  - Suppose you have 10,000 examples
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

Limitation of Accuracy

- Consider a 2-class problem
  - Suppose you have 10,000 examples
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0 accuracy = ?

Limitation of Accuracy

- Consider a 2-class problem
  - Suppose you have 10,000 examples
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10

- If model predicts everything to be class 0 accuracy = 9990/10,000 = 99,9% !!!

Cost Matrix

		PREDICTED CLASS	
ACTUAL CLASS	C(i | j)	**Class=Yes**	**Class=No**
	Class=Yes	C(Yes|Yes)	C(No|Yes)
	Class=No	C(Yes|No)	C(No|No)

C(i | j): Cost of misclassifying class **j** example as class **i**

Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
	C(i|j)	+	-
ACTUAL CLASS	+	-1	100
	-	1	0

Model M_1	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	150	40
	-	60	250

Accuracy = 80%

Cost = 3910

Model M_2	PREDICTED CLASS		
		+	-
ACTUAL CLASS	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

Cost vs Accuracy

Count	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a	b
	Class=No	c	d

Cost	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	p	q
	Class=No	q	p

Accuracy is proportional to cost if

1. C(Yes|No)=C(No|Yes) = q
2. C(Yes|Yes)=C(No|No) = p

$N = a + b + c + d$

$Accuracy = (a + d)/N$

$Cost = p\,(a + d) + q\,(b + c)$

$= p\,(a + d) + q\,(N - a - d)$

$= q\,N - (q - p)(a + d)$

$= N\,[q - (q-p) \times Accuracy]$

Cost-Sensitive Measures

$$\text{Precision (p)} = \frac{a}{a+c} = \frac{TP}{TP+FP}$$

$$\text{Recall (r)} = \frac{a}{a+b} = \frac{TP}{TP+FN}$$

$$\text{F - measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c} = \frac{2TP}{2TP+FP+FN}$$

		PREDICTED CLASS	
		Class= Yes	Class= No
ACTUAL CLASS	Class=Yes	a: TP	b: FN
	Class=No	c: FP	d: TN

- Precision is biased towards **C(Yes|Yes) & C(Yes|No)**
- Recall is biased towards **C(Yes|Yes) & C(No|Yes)**
- F-measure (weighted harmonic mean of precision and recall) is biased towards all except **C(No|No)**

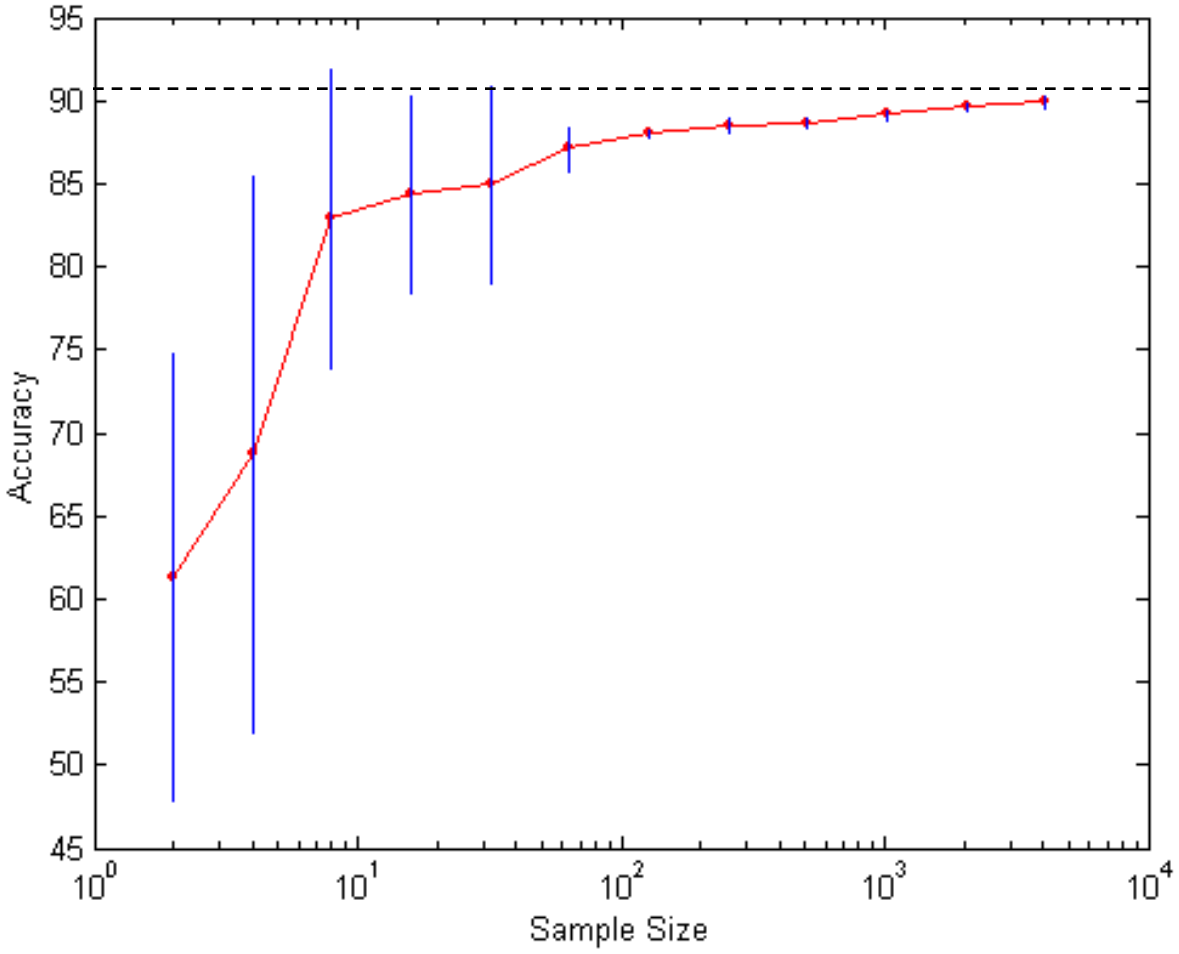
$$\text{Weighted Accuracy} = \frac{w_1 a + w_4 d}{w_1 a + w_2 b + w_3 c + w_4 d}$$

Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?

- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training and test sets

Learning Curve



- Learning curve shows how accuracy changes with varying sample size
- Requires a sampling schedule for creating learning curve

Effect of small sample size:

- Bias in the estimate
- Variance of estimate

Methods of Estimation

- **Holdout**
  - Reserve **2/3** for training and **1/3** for testing
- **Random subsampling**
  - Repeated holdout
- **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**
- **Bootstrap …**

Bootstrap

- Works well with small data sets
- Samples the given training examples uniformly *with replacement*
  - i.e., each time a example is selected, it is equally likely to be selected again and re-added to the training set
- **.632 bootstrap**
  - Suppose we are given a data set of d examples
  - The data set is sampled d times, with replacement, resulting in a training set of d new samples
  - The data examples that did not make it into the training set end up forming the test set
  - About 63.2% of the original data will end up in the bootstrap, and the remaining 36.8% will form the test set
  - Repeat the sampling procedure k times

ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms
- **ROC** curve plots **TPR** (on the **y**-axis) against **FPR** (on the **x**-axis)

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

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

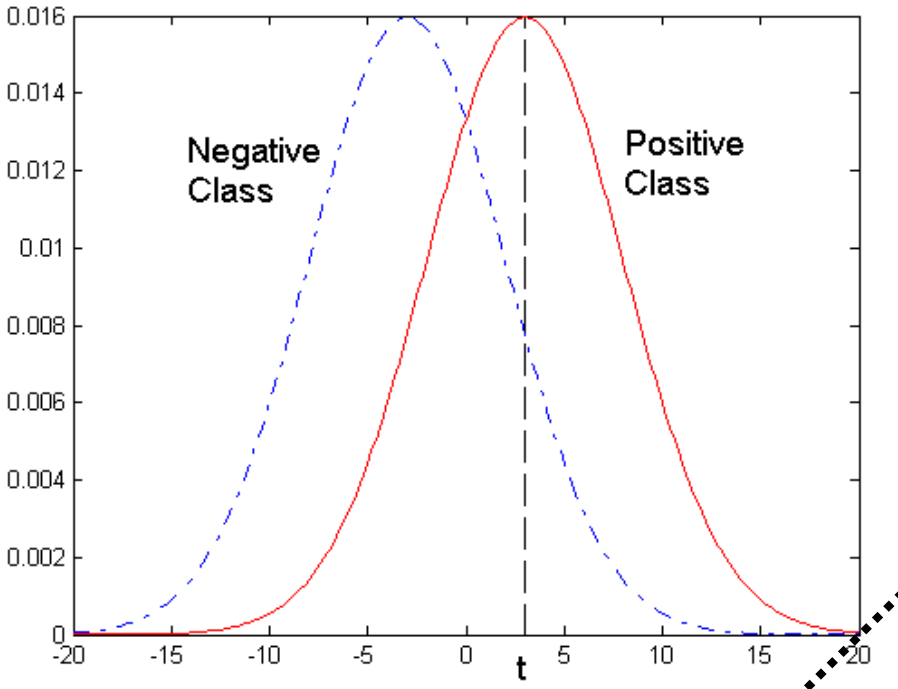
		PREDICTED CLASS	
		Yes	No
Actual	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

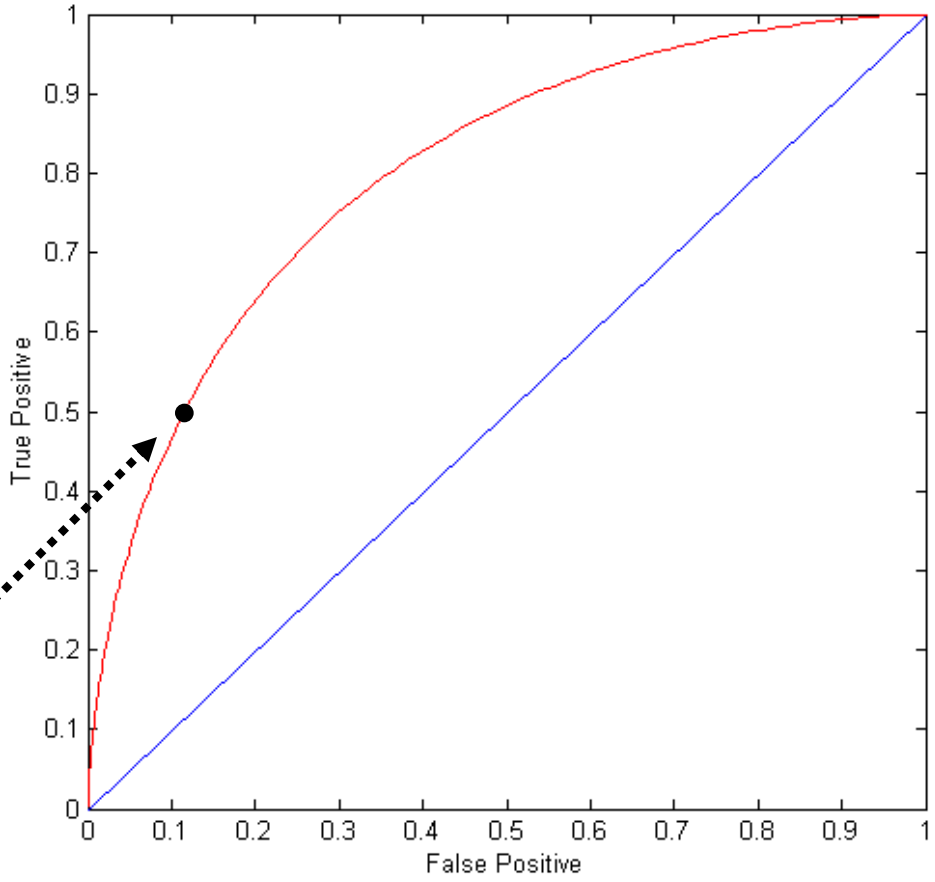
ROC (Receiver Operating Characteristic)

- Performance of each classifier represented as a point on the **ROC** curve
  - changing the threshold of algorithm, sample distribution, or cost matrix => changes the location of the point

ROC Curve

- **1**-dimensional data set containing **2** classes (***positive*** and ***negative***)
- any point located at **x > t** is classified as ***positive***





At threshold **t**:

TP=0.5, FN=0.5, FP=0.12, FN=0.88

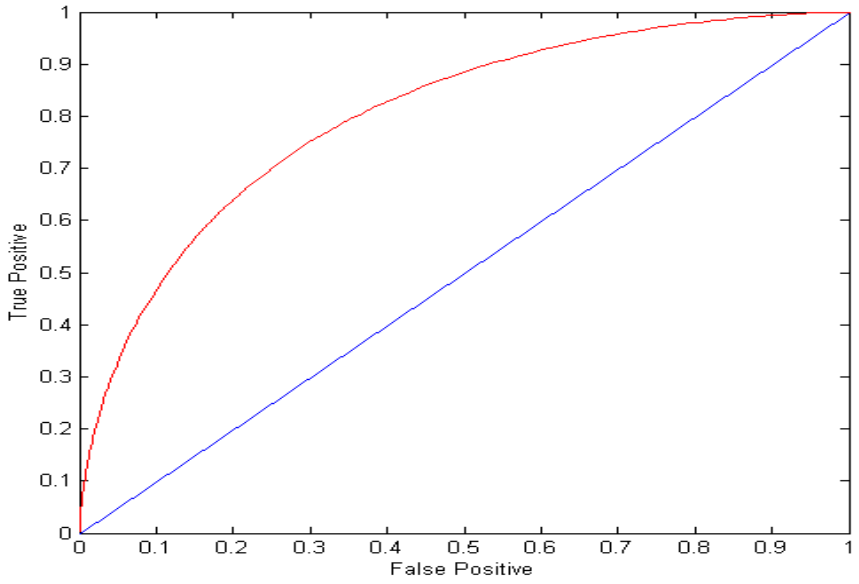
ROC Curve

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

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

(TP,FP):

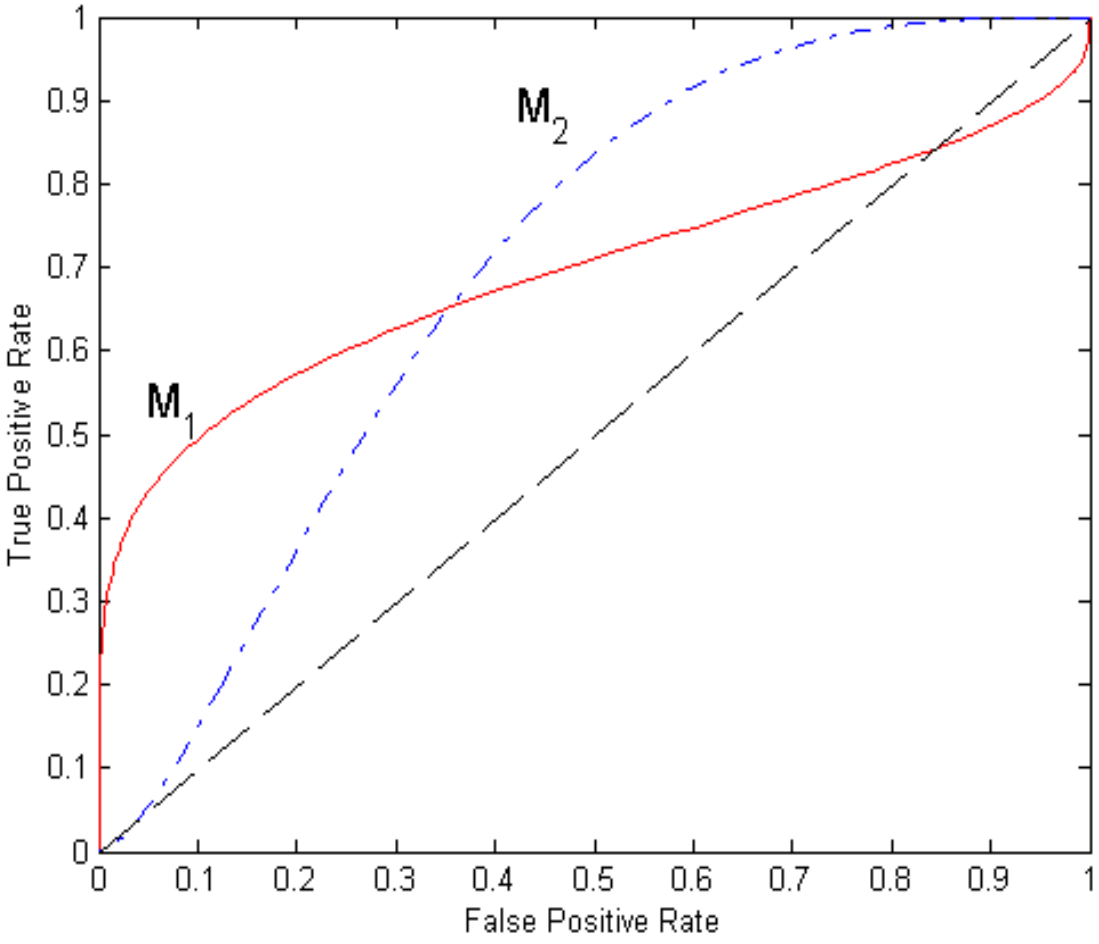
- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal



- Diagonal line:
  - Random guessing
  - Below diagonal line:
    - prediction is opposite of the true class

		PREDICTED CLASS	
		Yes	No
Actual	Yes	a (TP)	b (FN)
	No	c (FP)	d (TN)

Using ROC for Model Comparison



- No model consistently outperforms the other
  - M_1 is better for small FPR
  - M_2 is better for large FPR
- Area Under the ROC curve
  - Ideal: Area = 1
  - Random guess:
    - Area = 0.5

Today

- Forests (finishing up)
- Accuracy
- Evaluation methods
- ROC curves