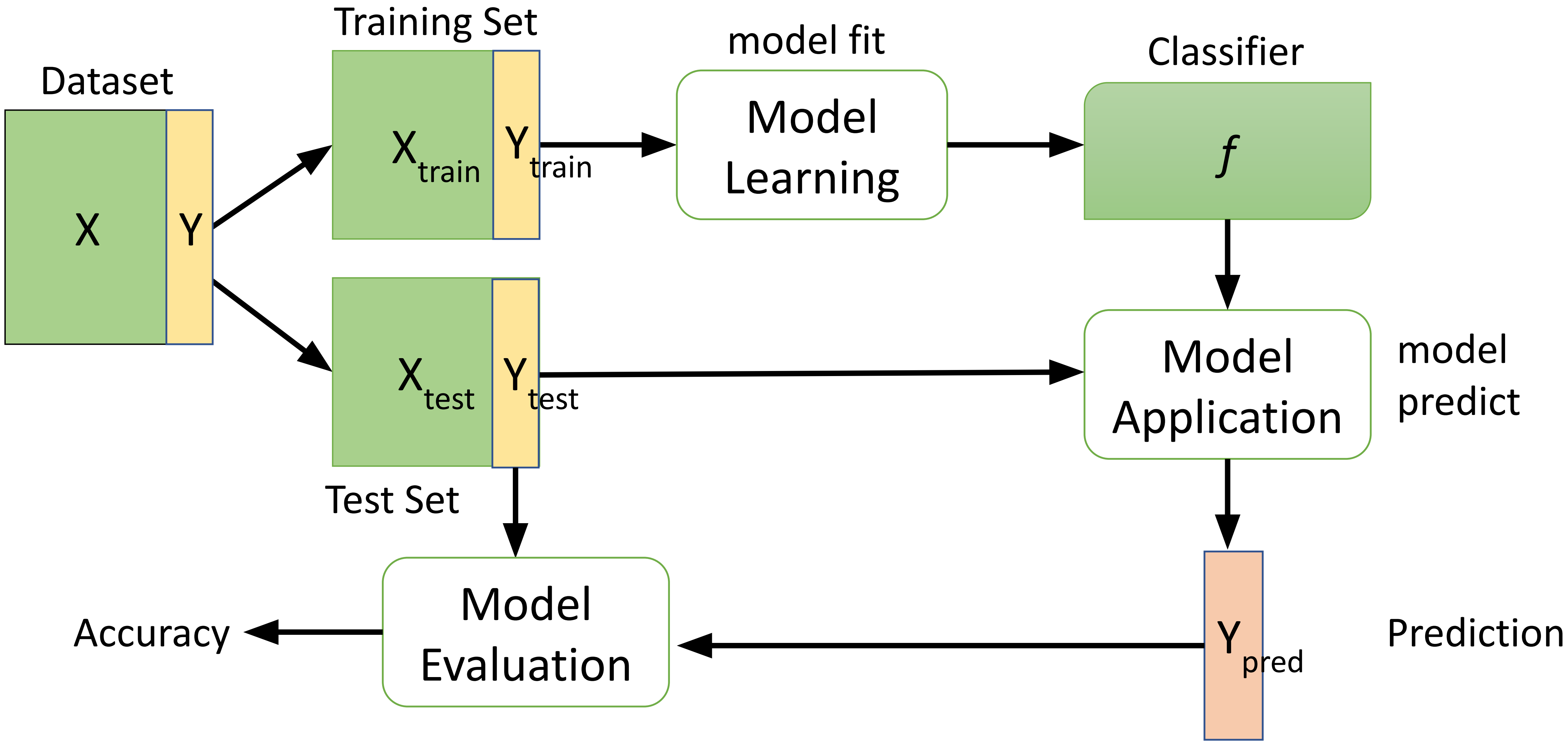
DATA MINING 1
Classification Model Evaluation

Dino Pedreschi, Riccardo Guidotti

Revisited slides from Lecture Notes for Chapter 3 "Introduction to Data Mining", 2nd Edition by Tan, Steinbach, Karpatne, Kumar

UNIVERSITÀ DI PISA

What is Classification?



Model Evaluation

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 among competing models?

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 among competing models?

Problem Setting

- Let suppose we have a vector y of actual/real class labels, i.e.,
- y = [0 0 0 1 1 1 0 1 0 1 0 1 1 1 0 0]

- Let name y' the vector returned by a trained model f, i.e.,
- y' = [0 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0]

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

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

Metrics for Performance Evaluation

- y = [0 0 0 1 1 1 0 1 0 1 0 1 1 1 0 0]
- y' = [0 0 1 1 1 0 0 1 0 1 1 1 0 0 0 0]
- TN FP FN TP

Metrics for Performance Evaluation…

	PREDICTED CLASS		
ACTUAL CLASS		Class=Yes	Class=No
	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
  - 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

Cost-Sensitive Measures

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

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

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

- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)

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

Cost Matrix

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

$N = a + b + c + d$

$\text{Accuracy} = (a + d)/N$

$$\begin{aligned}\text{Cost} &= p\,(a + d) + q\,(b + c) \\ &= p\,(a + d) + q\,(N - a - d) \\ &= q\,N - (q - p)(a + d) \\ &= N\,[q - (q-p) \times \text{Accuracy}]\end{aligned}$$

Binary vs Multiclass Evaluation

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	TP	FN
	Class=No	FP	TN

	PREDICTED CLASS			
		Class=A	Class=B	Class=C
ACTUAL CLASS	Class=A	TP-A		
	Class=B		TP-B	
	Class=C			TP-C

Accuracy = TP+TN / (TP+TN+FN+FP) = # correct / N

Accuracy = # correct / N = (TP-A + TP-B + TP-C) / N

Multiclass Evaluation

	PREDICTED CLASS			
		Class=A	Class=B	Class=C
ACTUAL CLASS	Class=A	TP-A	a	b
	Class=B	c	TP-B	d
	Class=C	e	f	TP-C

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

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

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

A	PREDICTED CLASS		
		Class=A	Class=Not A
ACTUAL CLASS	Class=A	TP-A	a + b
	Class=Not A	c + e	TP-B + TP-C + d + f

B	PREDICTED CLASS		
		Class=B	Class=Not B
ACTUAL CLASS	Class=B	TP-B	c + d
	Class=Not B	a + f	TP-A + TP-C + b + e

C	PREDICTED CLASS		
		Class=C	Class=Not C
ACTUAL CLASS	Class=C	TP-C	e + f
	Class=Not C	b + d	TP-A + TP-B + a + c

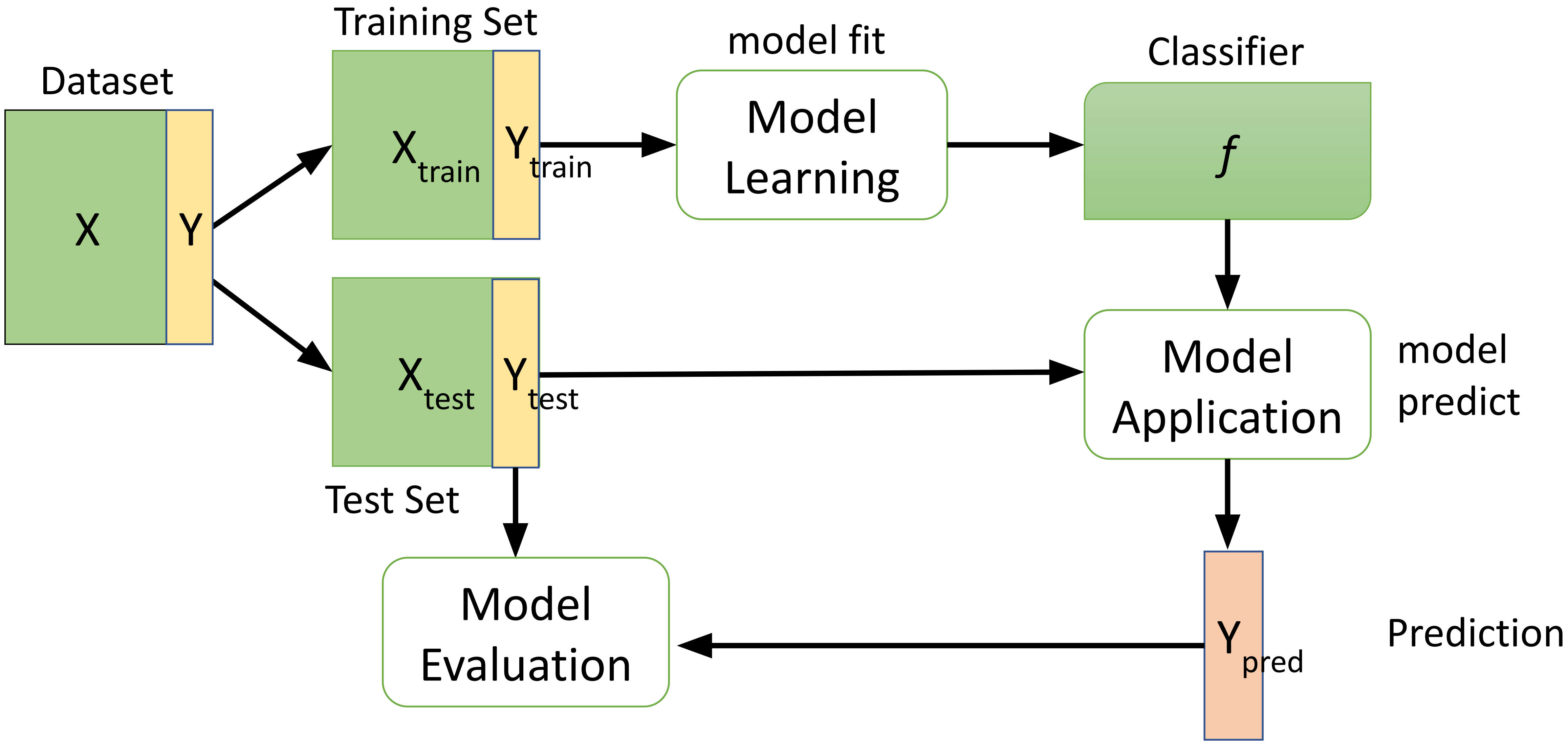
Model Evaluation

- 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 among competing models?

Methods for Evaluation



Parameter Tuning

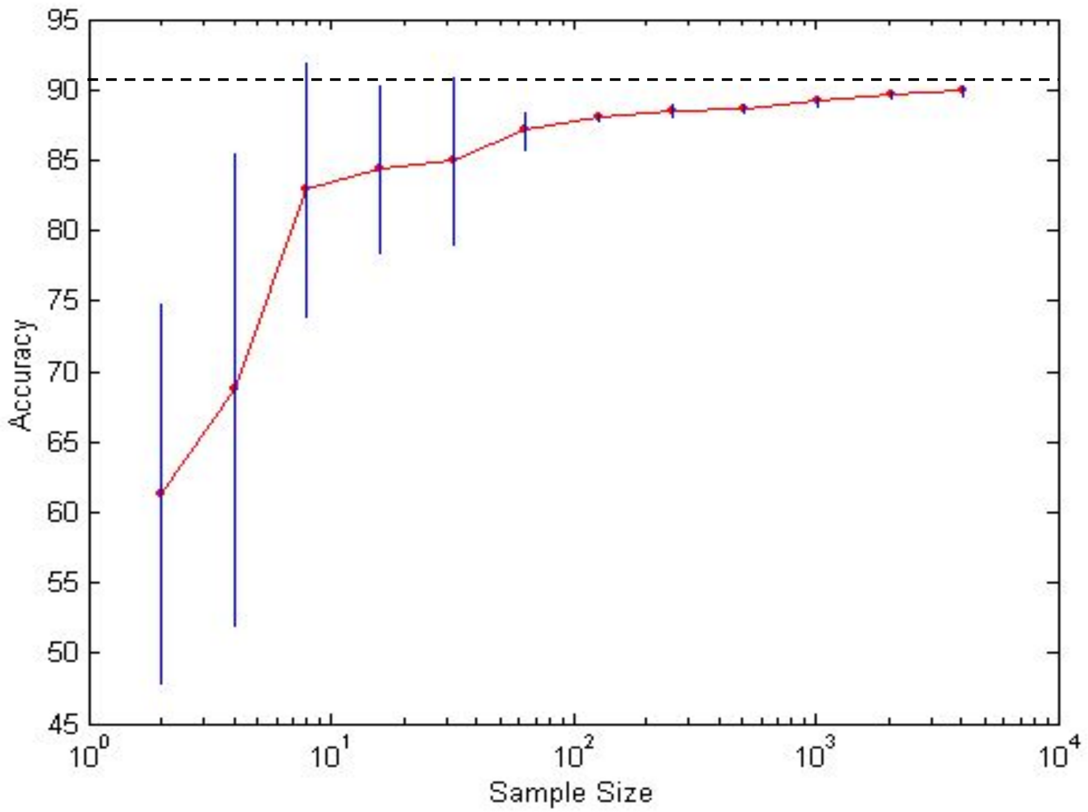
- It is important that the test data is not used in any way to create the classifier
- Some learning schemes operate in two stages:
  - **Stage 1**: builds the basic structure
  - **Stage 2**: optimizes parameter settings
  - **The test data can't be used for parameter tuning!**
  - Proper procedure uses three sets:
    - training data,
    - validation data,
    - test data
  - **Validation data is used to optimize parameters**
- Once evaluation is complete, all the data can be used to build the final classifier
- Generally, the larger the training data the better the classifier
- The larger the test data the more accurate the error estimate

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

1. How much a classification model benefits from adding more training data?
2. Does the model suffer from a variance error or a bias error?

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
- Stratified sampling
  - oversampling vs undersampling
- Bootstrap
  - Sampling with replacement

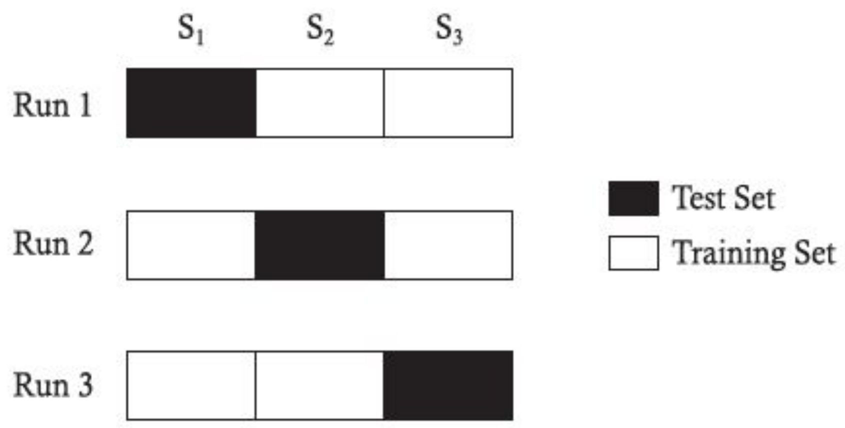
Holdout

- The holdout method reserves a certain amount for **testing** and uses the remainder for **training**
- Usually, **one third for testing**, the rest for training.
- Typical quantities are 60%-40%, 66%-34%, 70%-30%.
- For small or “unbalanced” datasets, **samples might not be representative**
  - For instance, few or none instances of some classes
- Stratified sample
  - **Balancing the data**
  - Make sure that each class is represented with approximately equal proportions in both subsets

Repeated Holdout

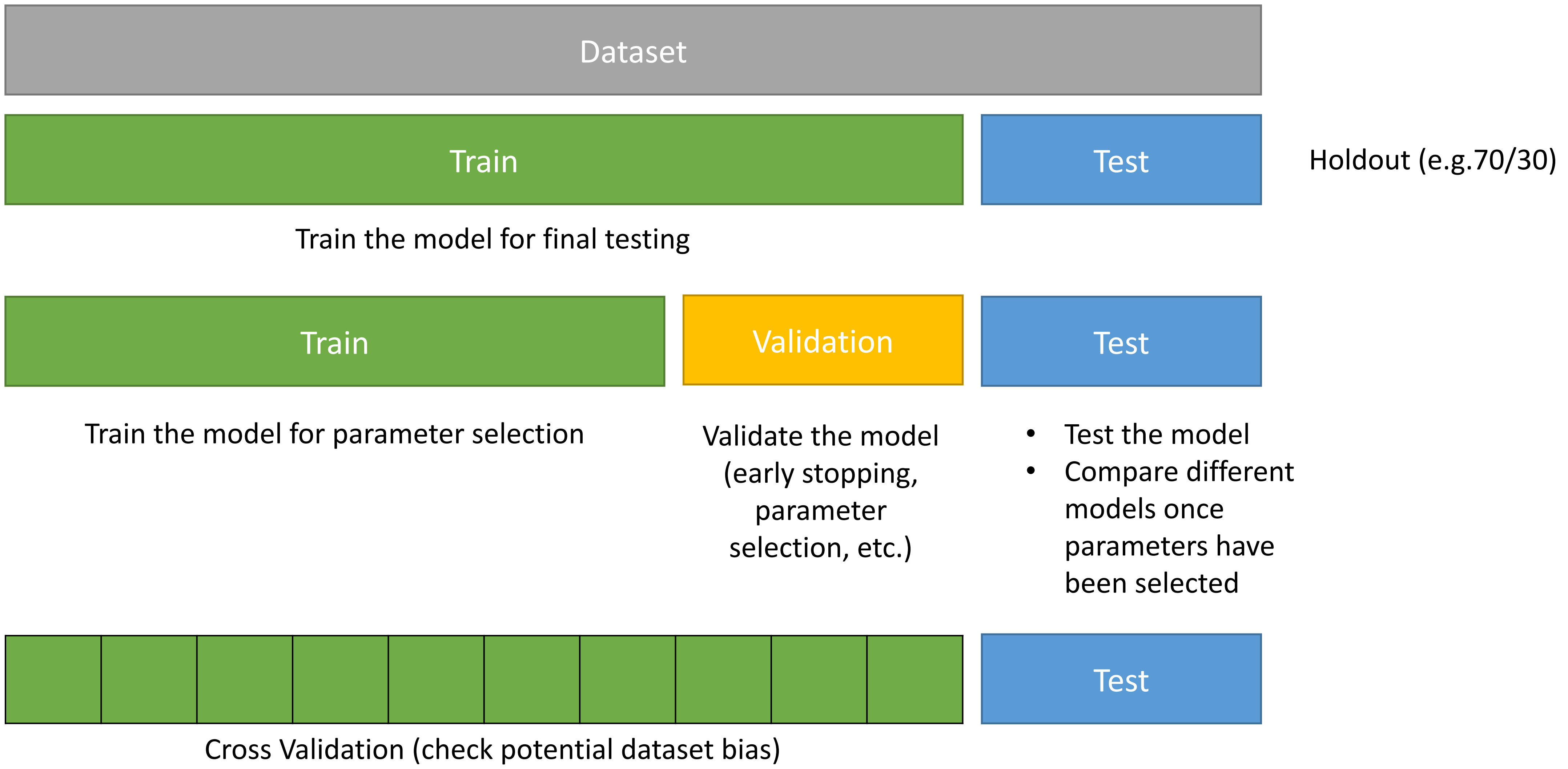
- Holdout estimate can be made more reliable by **repeating the process with different subsamples**
  - In each iteration, a certain proportion is **randomly selected for training** (possibly with stratification)
  - The error rates on the different iterations are **averaged** to yield an overall error rate
- This is called the **repeated holdout method**
- Still not optimum: the different test sets overlap

Cross Validation

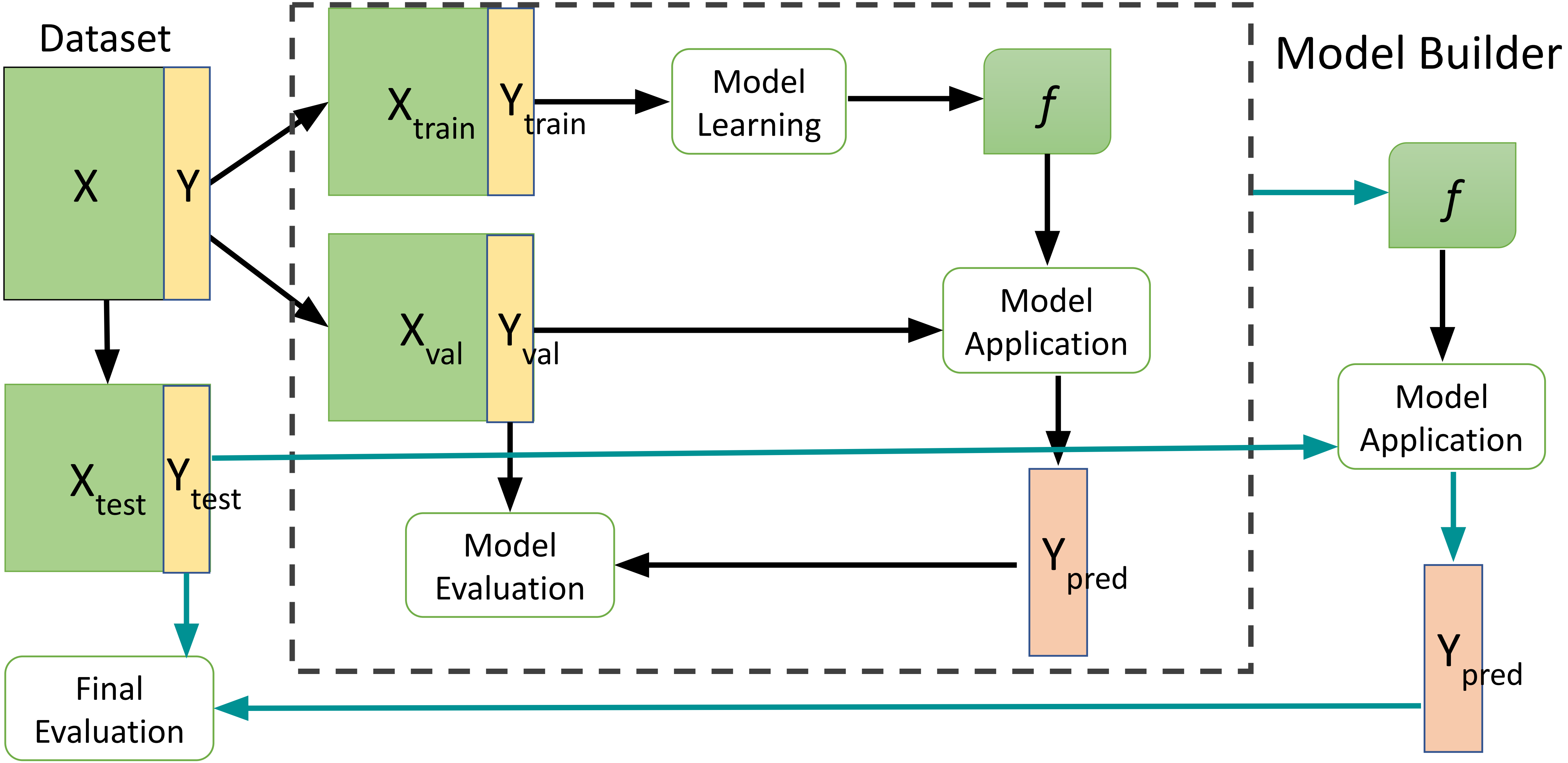


- Avoids overlapping test sets
  - **First step:** data is split into k subsets of equal size
  - **Second step:** each subset in turn is used for testing and the remainder for training
- This is called **k-fold cross-validation**
- Often the subsets are stratified before cross-validation is performed
- The **error estimates** are **averaged** to yield an overall error estimate
- **Even better:** repeated stratified cross-validation E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance)

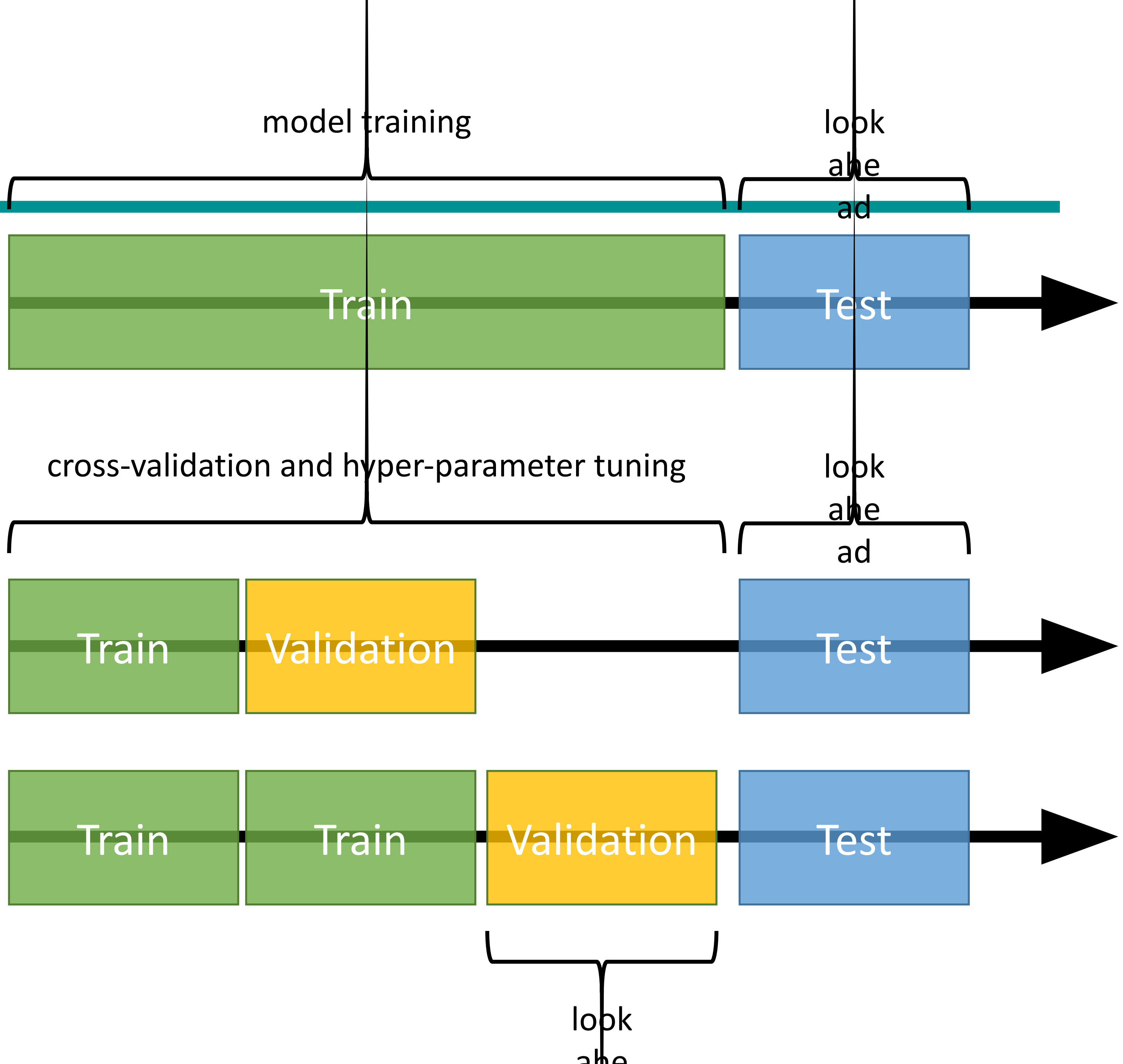
Data Partitioning



Evaluation: Training, Validation, Tests



Cross Validation with Time



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 among competing models?

ROC (Receiver Operating Characteristic)

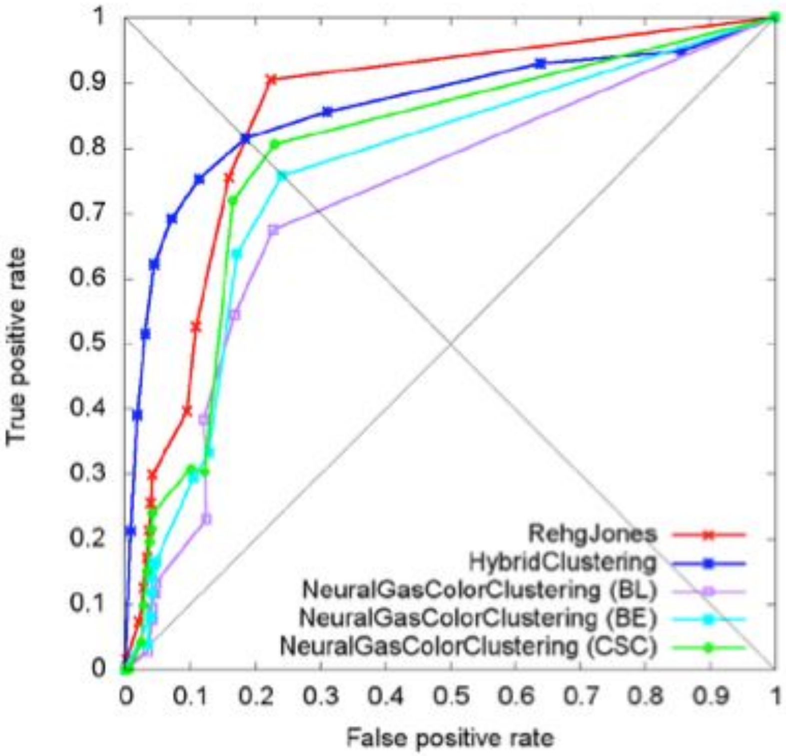
- Developed in 1950s for signal detection theory to analyze noisy signals
  - Characterize the trade-off between positive hits and false alarms

- ROC curve plots TPR (on the y-axis) against FPR (on the x-axis)

- **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

Receiver Operating Characteristic Curve

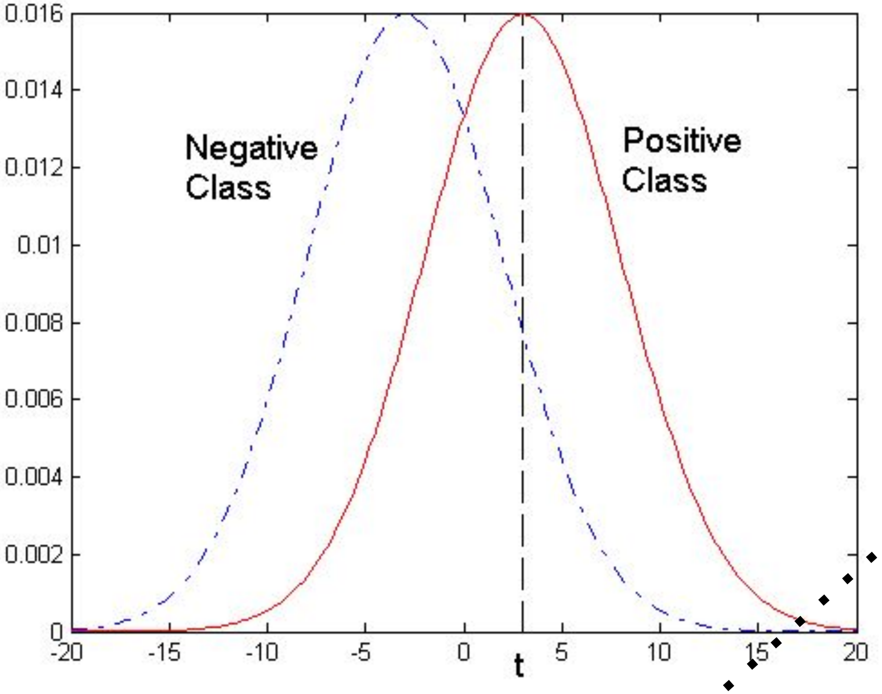
- It illustrates the ability of a binary classifier as its discrimination threshold THR is varied.
- The ***ROC*** curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various THR.
- The TPR = TP / (TP + FN) is also known as ***sensitivity***, ***recall*** or probability of detection.
- The FPR = FP / (TN + FP) is also known as probability of ***false alarm*** and can be calculated as (1 − specificity).

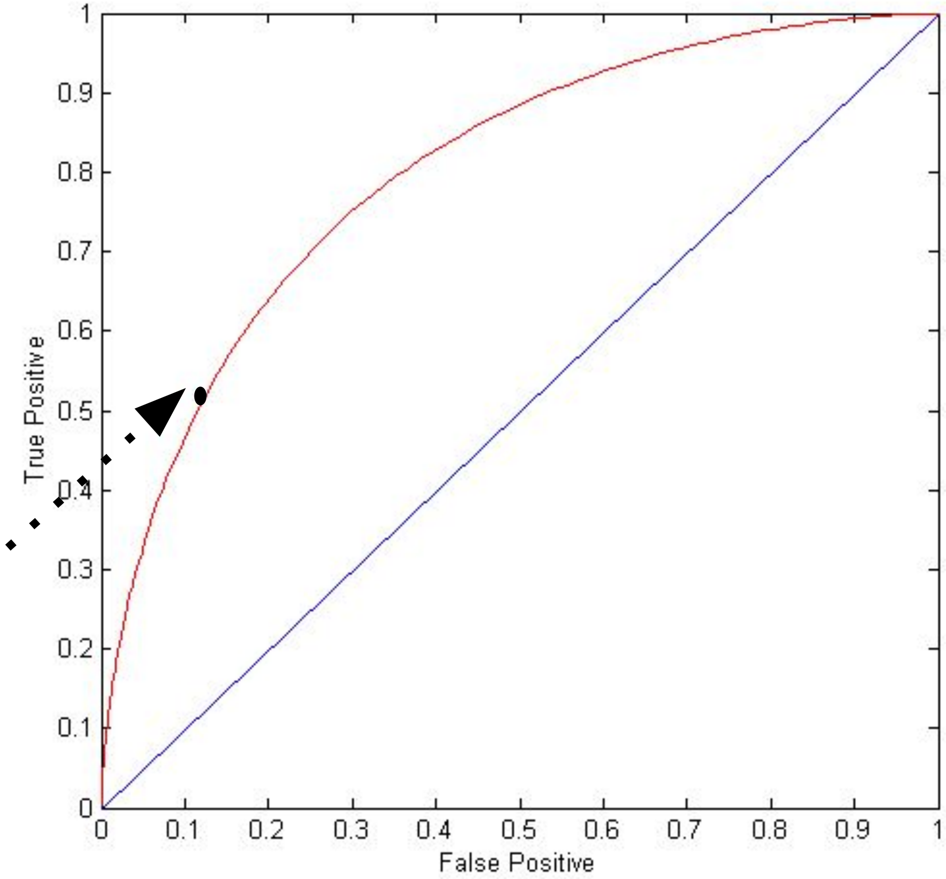


https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

ROC Curve

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





At threshold t:

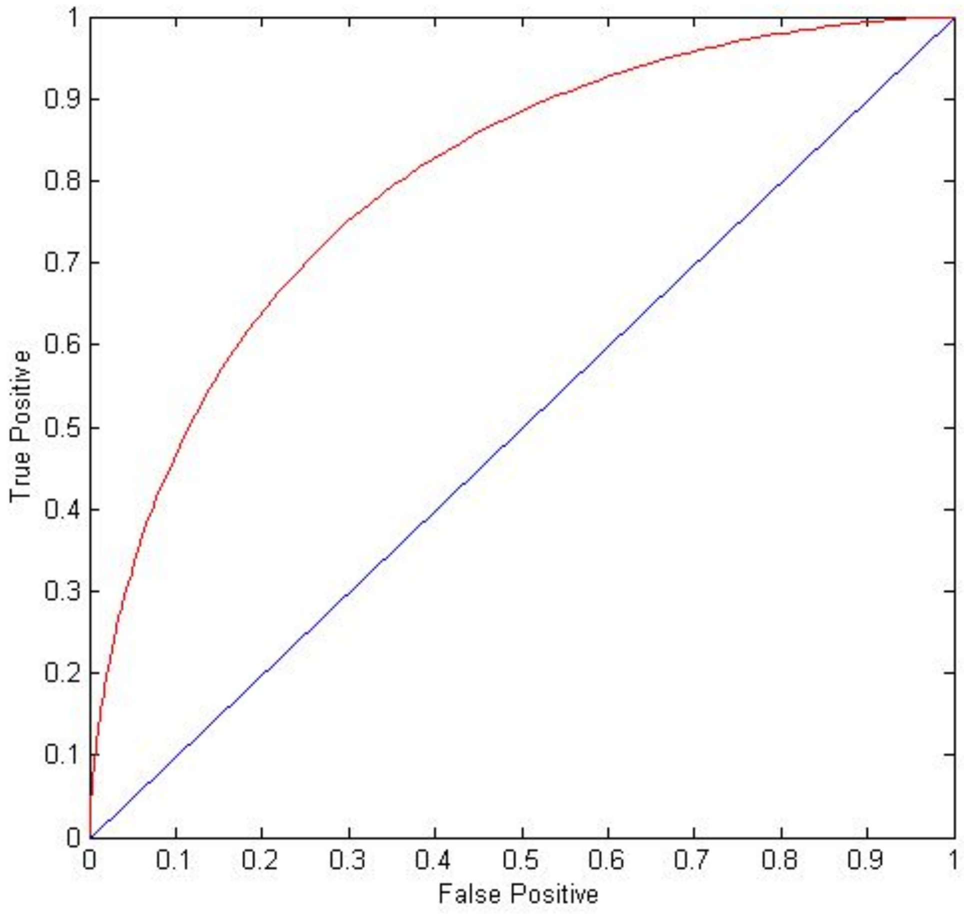
TP=0.5, FN=0.5, FP=0.12, FN=0.88

ROC Curve

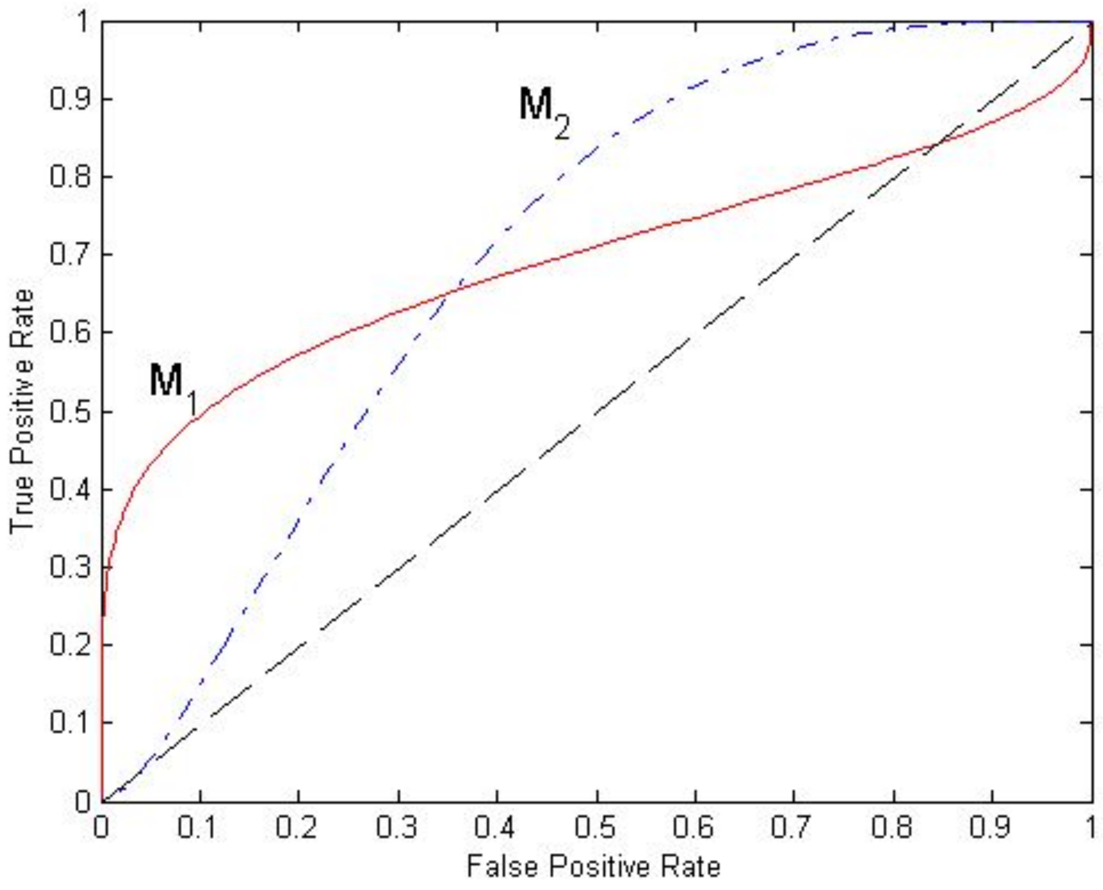
(TP,FP):

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

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



Using ROC for Model Comparison



- No model consistently outperform 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: Area = 0.5

How to Construct the ROC curve

Instance	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

- Use classifier that produces posterior probability for each test instance P(+|A)
- Sort the instances according to P(+|A) in decreasing order
- Apply threshold at each unique value of P(+|A)
- Count the number of TP, FP, TN, FN at each threshold
- TP rate, TPR = TP/(TP+FN)
- FP rate, FPR = FP/(FP + TN)

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

How to Construct the ROC curve

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP											
FP											
TN											
FN											
→ TPR											
→ FPR											

How to Construct the ROC curve

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5										
FP	5										
TN	0										
FN	0										
→ TPR	1										
→ FPR	1										

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

How to Construct the ROC curve

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4									
FP	5	5									
TN	0	0									
FN	0	1									
→ TPR	1	0.8									
→ FPR	1	1									

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

How to Construct the ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4								
FP	5	5	4								
TN	0	0	1								
FN	0	1	1								
→ TPR	1	0.8	0.8								
→ FPR	1	1	0.8								

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

How to Construct the ROC curve

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3							
FP	5	5	4	4							
TN	0	0	1	1							
FN	0	1	1	2							
→ TPR	1	0.8	0.8	0.6							
→ FPR	1	1	0.8	0.8							

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

How to Construct the ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3						
FP	5	5	4	4	3						
TN	0	0	1	1	2						
FN	0	1	1	2	2						
→ TPR	1	0.8	0.8	0.6	0.6						
→ FPR	1	1	0.8	0.8	0.6						

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

How to Construct the ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2			
FP	5	5	4	4	3	2	1	1			
TN	0	0	1	1	2	3	4	4			
FN	0	1	1	2	2	2	2	3			
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4			
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2			

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

How to Construct the ROC curve

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2		
FP	5	5	4	4	3	2	1	1	0		
TN	0	0	1	1	2	3	4	4	5		
FN	0	1	1	2	2	2	2	3	3		
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4		
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0		

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

$TPR = TP / (TP + FN)$

$FPR = FP / (TN + FP)$

How to Construct the ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	
FP	5	5	4	4	3	2	1	1	0	0	
TN	0	0	1	1	2	3	4	4	5	5	
FN	0	1	1	2	2	2	2	3	3	4	
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

How to Construct the ROC curve

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

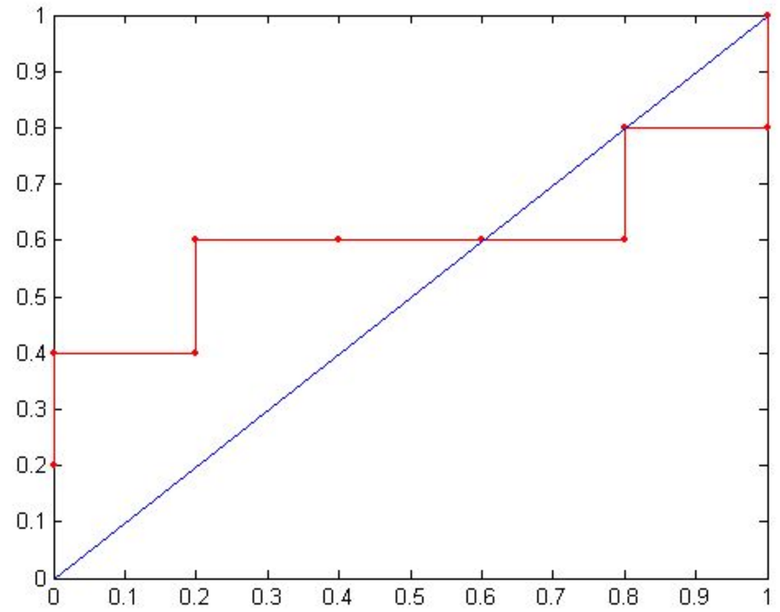
How to Construct the ROC curve

Class	+	-	+	-	-	-	+	-	+	+	
Threshold >=	0.25	0.43	0.53	0.76	0.85	0.85	0.85	0.87	0.93	0.95	1.00
TP	5	4	4	3	3	3	3	2	2	1	0
FP	5	5	4	4	3	2	1	1	0	0	0
TN	0	0	1	1	2	3	4	4	5	5	5
FN	0	1	1	2	2	2	2	3	3	4	5
→ TPR	1	0.8	0.8	0.6	0.6	0.6	0.6	0.4	0.4	0.2	0
→ FPR	1	1	0.8	0.8	0.6	0.4	0.2	0.2	0	0	0

Inst.	P(+|A)	True Class
1	0.95	+
2	0.93	+
3	0.87	-
4	0.85	-
5	0.85	-
6	0.85	+
7	0.76	-
8	0.53	+
9	0.43	-
10	0.25	+

TPR = TP / (TP + FN)
FPR = FP / (TN + FP)

ROC Curve:



Test of Significance

- Given two models:
  - Model M1: accuracy = 85%, tested on 30 instances
  - Model M2: accuracy = 75%, tested on 5000 instances

- Can we say M1 is better than M2?
  - How much confidence can we place on accuracy of M1 and M2?
  - Can the difference in performance measure be explained as a result of **random fluctuations** in the test set?

Confidence Interval for Accuracy

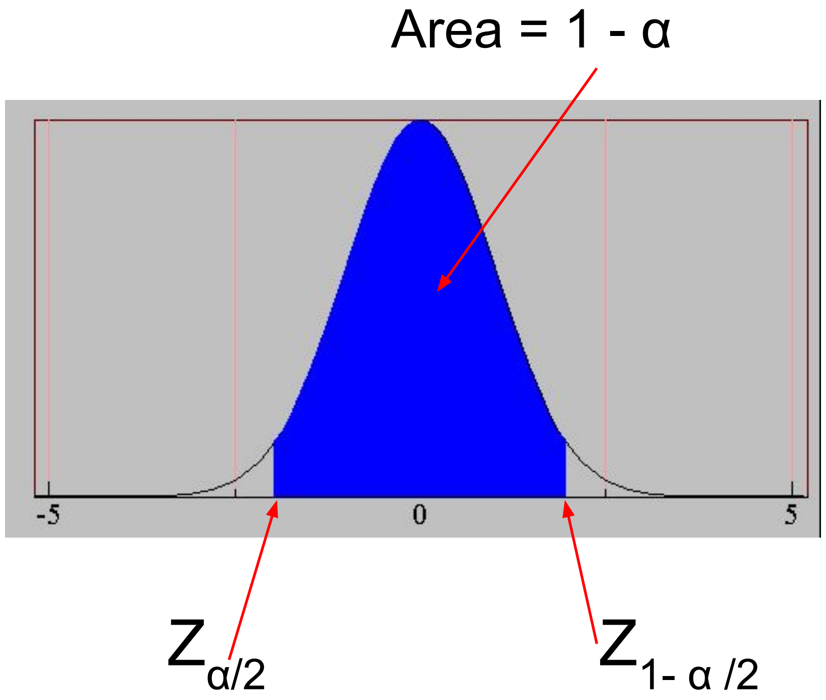
- Prediction can be regarded as a Bernoulli trial (binomial random experiment)
  - A Bernoulli trial has 2 possible outcomes
  - Possible outcomes for prediction: correct or wrong
  - Probability of success is constant
  - Collection of Bernoulli trials has a Binomial distribution:
    - $x \sim Bin(N, p)$ **x:** # of correct predictions, **N** trials, **p** constant prob.
    - e.g: Toss a fair coin 50 times, how many heads would turn up?
      Expected number of heads = $N \times p = 50 \times 0.5 = 25$

Given x (# of correct predictions) or equivalently, acc=x/N, and N (# of test instances)

Can we predict p (true accuracy of model)?

Confidence Interval for Accuracy

- For large test sets (N > 30),
  - acc has a normal distribution with mean p and variance p(1-p)/N



$$P\left(Z_{\alpha/2} < \frac{acc - p}{\sqrt{p(1-p)/N}} < Z_{1-\alpha/2}\right) = 1 - \alpha$$

- Confidence Interval for p:

$$p = \frac{2 \times N \times acc + Z_{\alpha/2}^2 \pm \sqrt{Z_{\alpha/2}^2 + 4 \times N \times acc - 4 \times N \times acc^2}}{2(N + Z_{\alpha/2}^2)}$$

Confidence Interval for Accuracy

- Consider a model that produces an accuracy of 80% when evaluated on 100 test instances:
  - N=100, acc = 0.8
  - Let 1-α = 0.95 (95% confidence)
  - **Which is the confidence interval?**
  - From probability table, $Z_{\alpha/2}$=1.96

1-α	Z
0.99	2.58
0.98	2.33
0.95	1.96
0.90	1.65

N	50	100	500	1000	5000
p(lower)	0.670	0.711	0.763	0.774	0.789
p(upper)	0.888	0.866	0.833	0.824	0.811

Comparing Performance of 2 Models

- Given two models, say M1 and M2, which is better?
  - M1 is tested on D1 (size=n1), found error rate = e_1
  - M2 is tested on D2 (size=n2), found error rate = e_2
  - Assume D1 and D2 are independent
  - If n1 and n2 are sufficiently large, then

$$e_1 \sim N(\mu_1, \sigma_1)$$
$$e_2 \sim N(\mu_2, \sigma_2)$$

  - Approximate variance of error rates:

$$\hat{\sigma}_i = \frac{e_i(1-e_i)}{n_i}$$

Comparing Performance of 2 Models

- To test if performance difference is statistically significant: $d = e_1 - e_2$
  - $d \sim N(d_t, \sigma_t)$ where d_t is the true difference
  - Since D1 and D2 are independent, their variance adds up:

$$\sigma_t^2 = \sigma_1^2 + \sigma_2^2 \cong \hat{\sigma}_1^2 + \hat{\sigma}_2^2$$

$$= \frac{e1(1-e1)}{n1} + \frac{e2(1-e2)}{n2}$$

  - It can be shown at (1-α) confidence level,

$$d_t = d \pm Z_{\alpha/2} \hat{\sigma}_t$$

An Illustrative Example

- Given: M1: n1 = 30, e1 = 0.15
  M2: n2 = 5000, e2 = 0.25

- d = |e2 – e1| = 0.1 (2-sided test to check: dt = 0 or dt <> 0)

$$\hat{\sigma}_d^2 = \frac{0.15(1-0.15)}{30} + \frac{0.25(1-0.25)}{5000} = 0.0043$$

- At 95% confidence level, $Z_{\alpha/2}$=1.96

$$d_t = 0.100 \pm 1.96 \times \sqrt{0.0043} = 0.100 \pm 0.128$$

=> Interval contains 0 => difference may not be statistically significant

Comparing Performance of 2 Algorithms

- Each learning algorithm may produce k models:
  - L1 may produce M11 , M12, …, M1k
  - L2 may produce M21 , M22, …, M2k
- If models are generated on the same test sets D1,D2, …, Dk (e.g., via cross-validation)
  - For each set: compute $d_j = e_{1j} – e_{2j}$
  - d_j has mean d_t and variance σ_t^2
  - Estimate:

$$\hat{\sigma}_t^2 = \frac{\sum_{j=1}^{k}(d_j - \overline{d})^2}{k(k-1)}$$

$$d_t = \overline{d} \pm t_{1-\alpha,k-1}\hat{\sigma}_t$$

Lift Chart

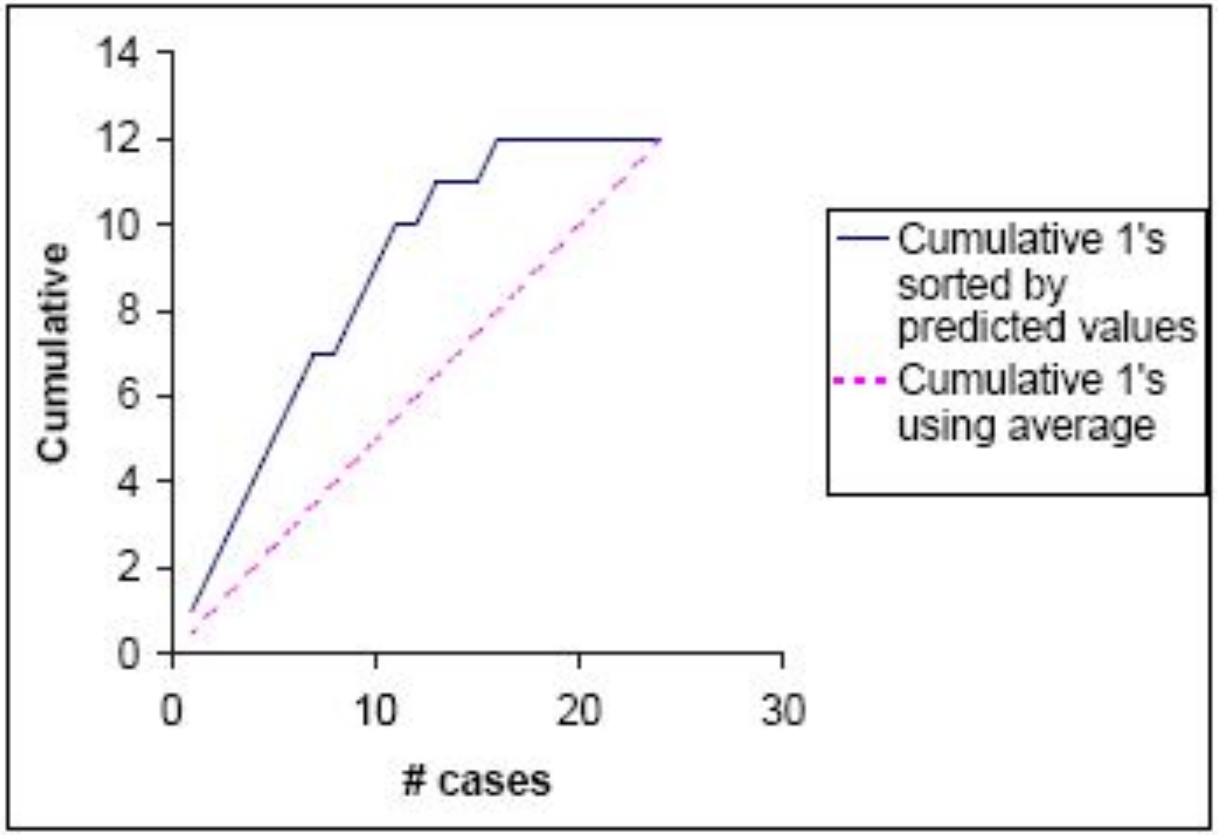
http://www2.cs.uregina.ca/~dbd/cs831/notes/lift_chart/lift_chart.html
http://mlwiki.org/index.php/Cumulative_Gain_Chart

- The lift curve is a popular technique in direct marketing.
- The input is a dataset that has been "scored'' by appending to each case the estimated probability that it will belong to a given class.
- The cumulative ***lift chart*** (also called ***gains chart***) is constructed with the cumulative number of cases (descending order of probability) on the x-axis and the cumulative number of true positives on the y-axis.
- The dashed line is a reference line. For any given number of cases (the x-axis value), it represents the expected number of positives we would predict if we did not have a model but simply selected cases at random. It provides a benchmark against which we can see performance of the model.

Notice: "Lift chart" is a rather general term, often used to identify also other kinds of plots. Don't get confused!

Lift Chart – Example

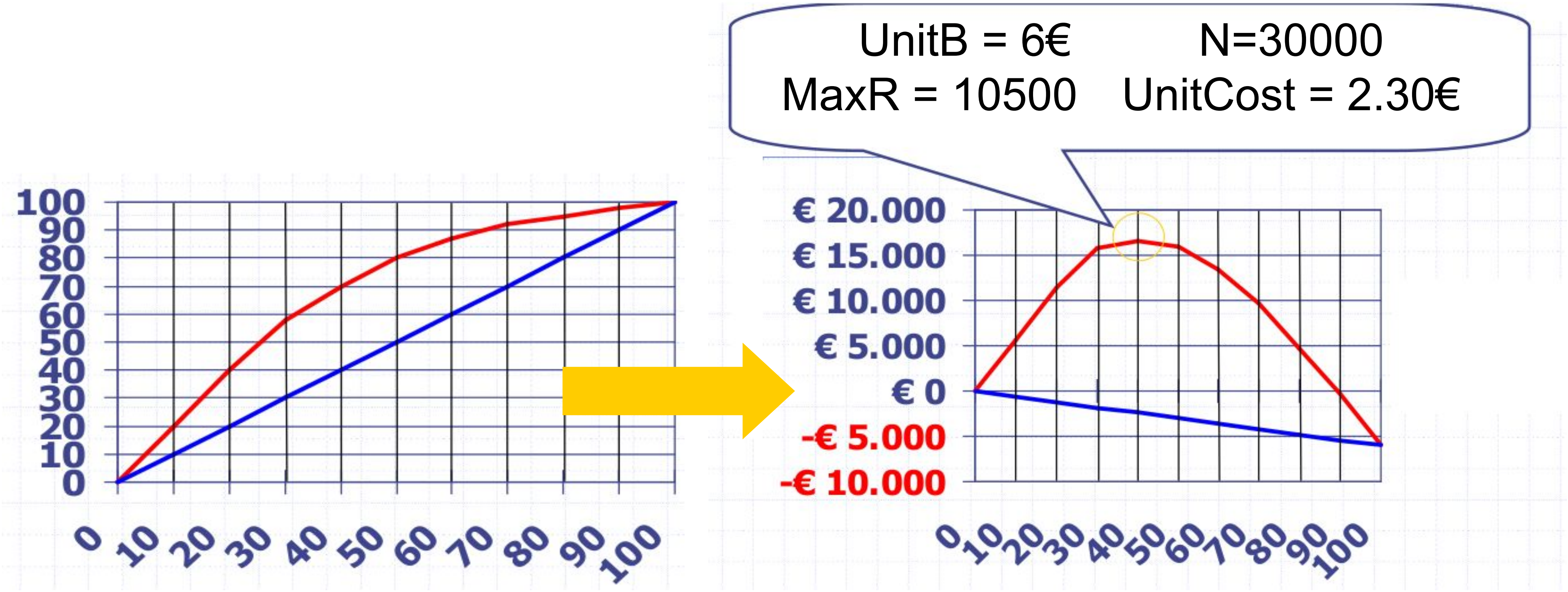
Serial no.	Predicted prob of 1	Actual Class	Cumulative Actual class
1	0.995976726	1	1
2	0.987533139	1	2
3	0.984456382	1	3
4	0.980439587	1	4
5	0.948110638	1	5
6	0.889297203	1	6
7	0.847631864	1	7
8	0.762806287	0	7
9	0.706991915	1	8
10	0.680754087	1	9
11	0.656343749	1	10
12	0.622419543	0	10
13	0.505506928	1	11
14	0.47134045	0	11
15	0.337117362	0	11
16	0.21796781	1	12
17	0.199240432	0	12
18	0.149482655	0	12
19	0.047962588	0	12
20	0.038341401	0	12
21	0.024850999	0	12
22	0.021806029	0	12
23	0.016129906	0	12
24	0.003559986	0	12



Lift Chart – Application Example

- From Lift chart we can easily derive an "economical value" plot, e.g. in target marketing.
- Given our predictive model, how many customers should we target to maximize income?
- Profit = UnitB*MaxR*Lift(X) - UnitCost*N*X/100
- UnitB = unit benefit, UnitCost = unit postal cost
- N = total customers
- MaxR = expected potential respondents in all population (N)
- Lift(X) = lift chart value for X, in [0,..,1]

Lift Chart – Application Example



References

- Chapter 3. Classification: Basic Concepts and Techniques.