Business Intelligence and Data Management

academic year 2019-2020

Dr. Ekaterini Ioannou, Department of Management, University of Tilburg

TILBURG UNIVERSITY

Lecture 8

topic: Performance Measures

material: Chapters 5 (book "Data Mining for Business Intelligence")

If you can't measure it, you can't improve it!

Agenda

Date		Lecture contents	Lecturer	Lab topics	Test
Jan-27	1	Intro. to BI+ Data Management	Caron		
Jan-30				SQL-1	1
Jan-28	2	Data warehousing	Caron		
Feb-06				SQL-2	2
Feb-03	3	OLAP business databases & dashboard	Caron		
Feb-13				SQL-3 & OLAP	3a & 3b
Feb-10	4	Data mining introduction	Ioannou		
	5	Regression models	Ioannou		
Feb-17	6	Naïve Bayes	Ioannou		
	7	k nearest neighbors	Ioannou		
Feb-20				Bayes & neighbors	4
Feb-27	8	Performance measures	Ioannou		
Mar-02	9	Decision trees	Ioannou		
Mar-05				Dec. trees	5
Mar-09	10	Association rules	Ioannou		
Mar-11,12&13				Ass. Rules	6
Mar-16	11	Clustering (+20 mins exam preparation)	Ioannou		
Mar-19				Clustering	7

Why shall we evaluate?

- Concerns both researchers and practitioners
- It allows you to convince others that your work is meaningful
  - Examples: clients, peers, funding agencies, company VPs, investors, teachers, …
- Without a strong evaluation, your idea is likely to be rejected, code would not be deployed
- Empirical evaluation helps guide meaningful research and development directions

Evaluation issues

"If you cannot measure it, you cannot improve it"

- How to evaluate the performance of a model?
- How reliable are the predicted results?
- How to obtain reliable estimates?
- How to compare the relative performance among competing models?
- How to select among models that have equal performance?

Evaluation issues

- Various models, e.g.:
  - Classify instances
  - Predicting class probability
  - Predicting numeric rather than nominal values

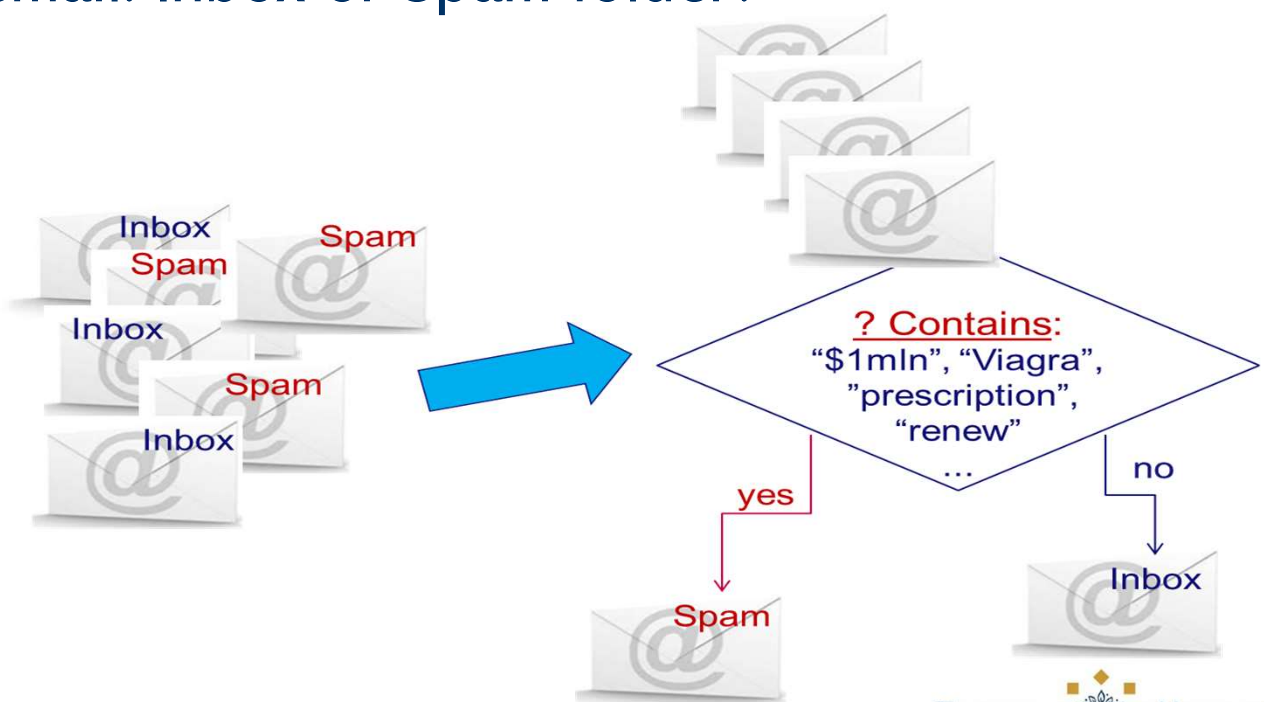
→ Not a single methodology for the evaluation!

→ Next slides: various possible evaluation measures

Classification

Classify e-mails to spam vs. ham

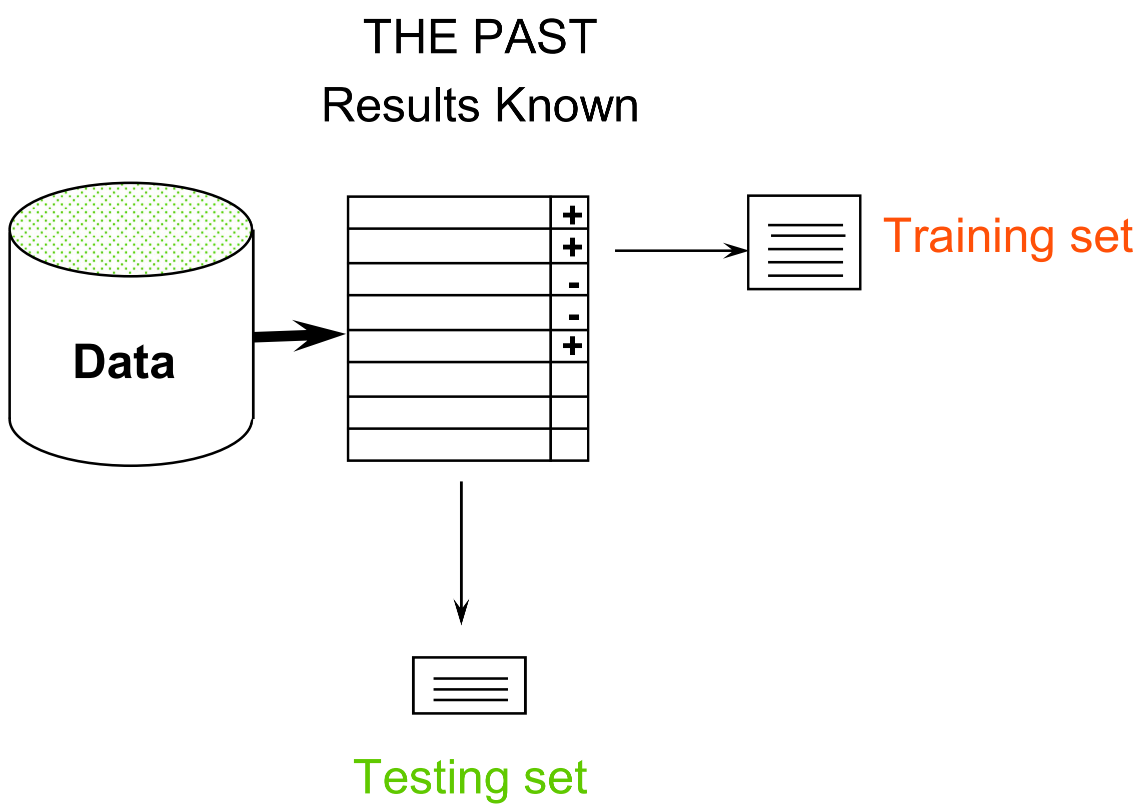
- Given (labeled) examples of both document types
- Train a classifier to discriminate between these two
- During operation, use classifier to select destination folder for new email: Inbox or Spam folder?



Classification

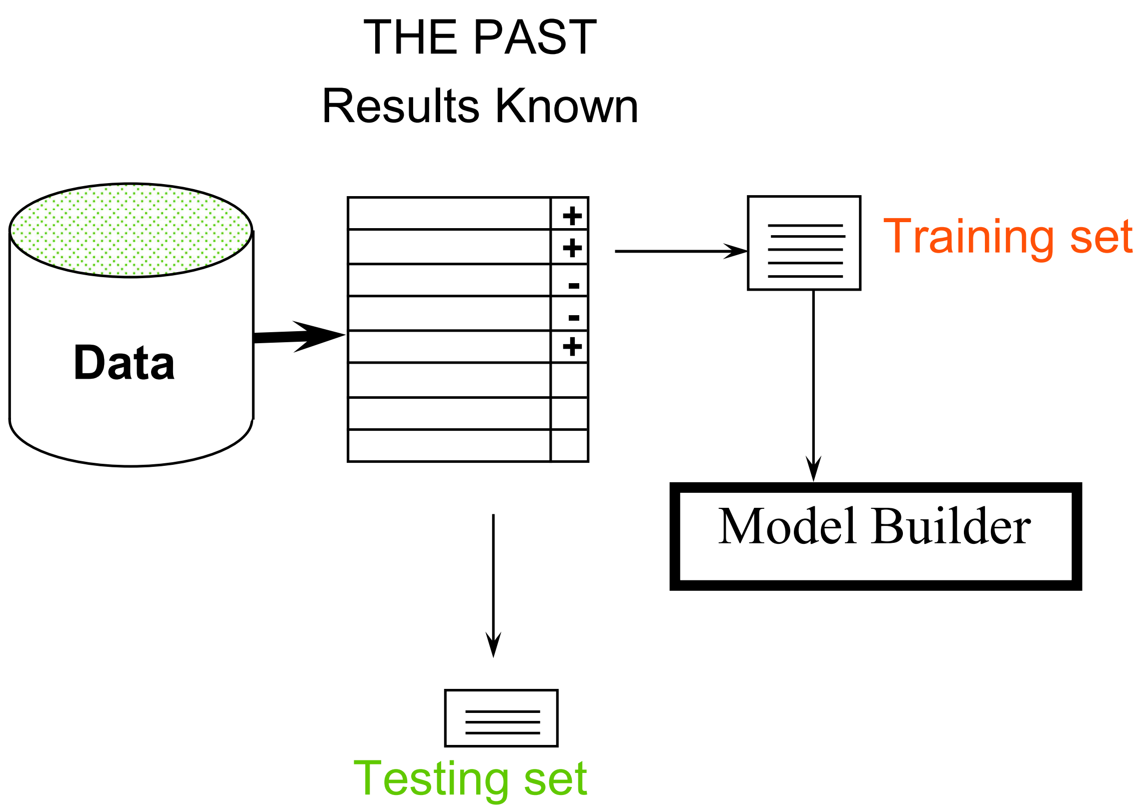
Classification Step 1:

Split data into train and test sets



Classification

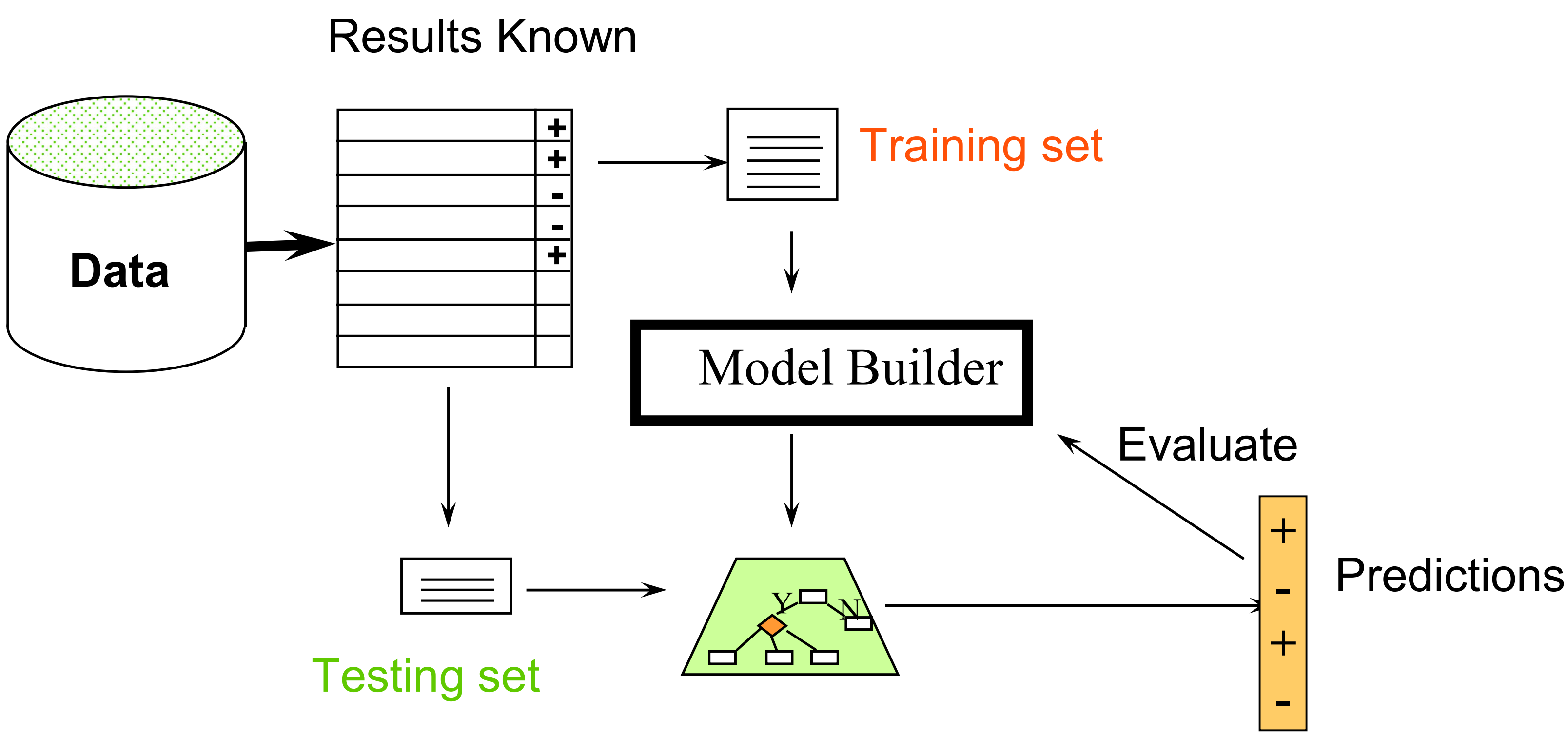
Classification Step 2:
Build a model on a training set



Classification

Classification Step 3:

Evaluate on test set (Re-train?)

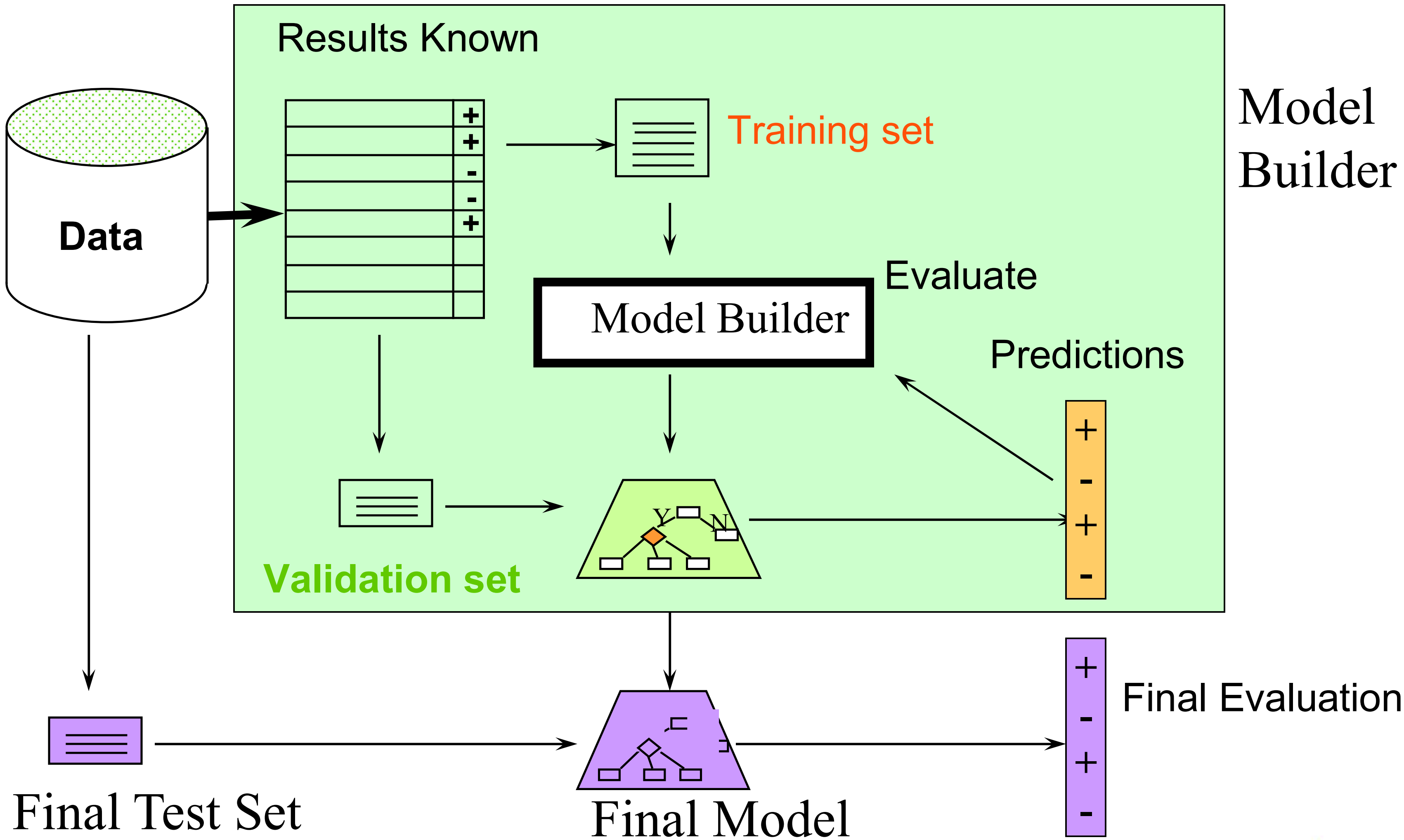


Note on parameter tuning

- It is important that the test data is not used in any way to create the classifier
- Some training schemes operate in two stages:
  - Stage 1: build the basic structure
  - Stage 2: optimize parameter settings

- The test data cannot be used for parameter tuning!
- Proper procedure uses three sets: training data, validation data, and test data
  - Validation data is used to optimize parameters

Classification



Making the most of the data

- Once evaluation is completed, all the data can be used to build the final classifier
- Generally, the larger the training data the better the classifier (but returns diminish)
- The larger the test data the more accurate the error estimate

Training and Validation Performance

- Errors that are based on the training set tell us about model fit
- Errors that are based on the validation set measure the model's ability to predict new data
  - Prediction errors that measure predictive performance

What is the expected relationship between training errors and validation errors?

a) Training errors are less than the validation errors
b) Validation errors are less than the training errors
c) Cannot predict the relationship
d) Exactly the same

Types of outcomes

- Building models using training data is called Supervised Learning
- Interested in predicting the outcome variable for new records
- Three main types of outcomes
  a. Predicted numerical value, e.g., house price
  b. Predicted class membership, e.g., cancer or not
  c. Probability of class membership (for categorical outcome variable), e.g., Naive Bayes

Evaluating Predictive Performance

i.e., numerical (continuous) variables

Generating numeric predictions

- Interested in models that have high predictive accuracy when applied to new records

- Models are trained on the training data
- Applied to the validation data and
- Measures of accuracy then use the prediction errors on that validation set

Prediction Accuracy measures

- Prediction error for record i is the difference between its actual outcome value y_i and its predicted outcome value $\hat{y}_i$:

$$e_i = y_i - \hat{y}_i$$

- Several popular numerical measures

Prediction Accuracy measures

Mean absolute error/deviation (MAE)

- Gives the magnitude of the average absolute error

$$\frac{1}{n}\sum_{i=1}^{n} |\, \mathrm{e}_i \,|$$

Mean error

- Same as MAE but no absolute value

$$\frac{1}{n}\sum_{i=1}^{n} \mathrm{e}_i$$

- Negative errors cancel positive of same magnitude
- Indication of whether predictions are on average over-/under-predicting the outcome variable

Prediction Accuracy measures

Mean absolute error/deviation (MAE)

Mean error

Mean percentage error (MPE)

- Percentage score of how predictions deviate from the actual values (on average)
- Takes into account the direction of the error

$$100\,\frac{1}{n}\sum_{i=1}^{n}\frac{\mathrm{e}_i}{\mathrm{y}_i}$$

Mean absolute percentage error (MAPE)

- Percentage score of how predictions deviate (on average) from the actual values

$$100\,\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\mathrm{e}_i}{\mathrm{y}_i}\right|$$

Prediction Accuracy measures

Mean absolute error/deviation (MAE)

Mean error

Mean percentage error (MPE)

Mean absolute percentage error (MAPE)

Root mean squared error

- Intuition: (normalized) distance between the vector of predicted values and the vector of actual value
- Same units as the outcome variable

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} \mathrm{e}_i^{\,2}}$$

Lift Chart

- Graphical way to assess predictive performance
- In some applications, we are not interested in predicting the outcome of value of each new record
- But the goal is to search for a subset of records that gives the highest cumulative predicted values

- Compares the model's predictive performance to a baseline model that has no predictors

Lift chart

- In practice, costs are rarely known
- Decisions are usually made by comparing possible scenarios
- Example: promotional mailout to 1,000,000 households
  - Mail to all with response rate being 0.1% (1,000)
  - Consider a data mining tool that can identify a subset of 100,000 most promising households with response rate 0.4% (400)
  - Responses are 40% BUT cost is 10%

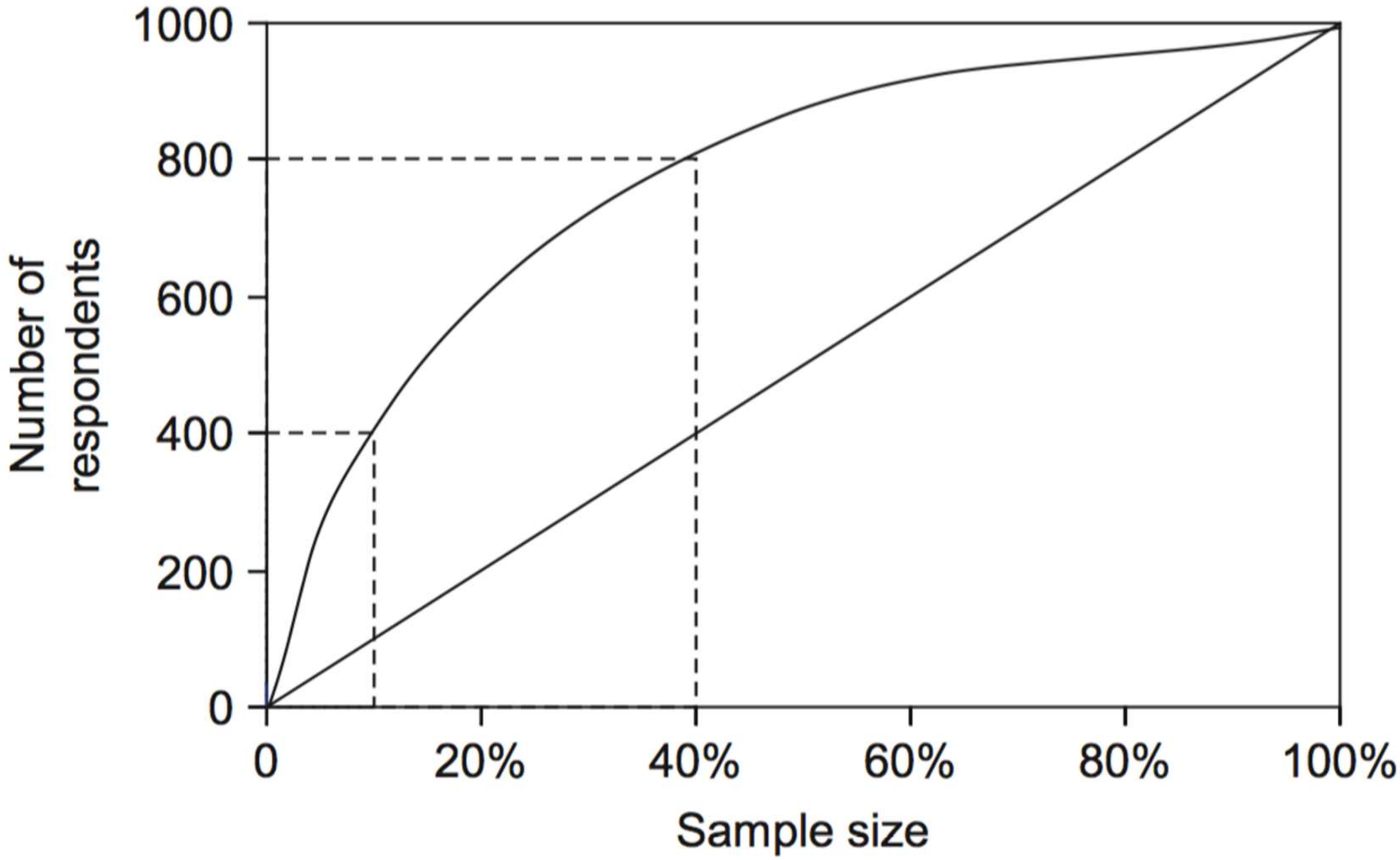
  ➔ It might pay off to restrict to these 100,000
- The increase in response rate is called lift factor
- A lift chart allows a visual comparison

Generating a lift chart

Rank	Predicted	Actual
1	0.95	*Yes*
2	0.93	*Yes*
3	0.93	*No*
4	0.88	*Yes*
5	0.86	*Yes*
6	0.85	*Yes*
7	0.82	*Yes*
8	0.80	*Yes*
9	0.80	*No*
10	0.79	*Yes*

- Take sample data set
- Apply the selected model
- Sort according to predicted probability of a yes response, i.e.,
  - First instance is the one the model thinks is more likely a yes
  - Next instance is the next most likely
  - Etc.
- Intuition: more yes at the beginning of the list
- x-axis is sample percentage, i.e., 20% from the start of the list
- y-axis is response number, i.e., percentage where the model correctly predicts the positive class

A hypothetical lift chart



40% of responses
for 10% of cost

80% of responses
for 40% of cost

Judging Classifier Performance

i.e., categorical variables

Misclassification error

- Perfect classifier makes no errors!
- But the real world has "noise" and not all the information needed to classify records precisely
- Cannot construct perfect classifiers

- Misclassification is when a record belongs to one class but the model classifies it as a member of a different class
- A natural criterion for judging the performance of a classifier is the probability of making a misclassification error

Confusion / Classification matrix

- A matrix that summarizes the correct and incorrect classifications that a classifier produced for a given dataset
- Rows and columns correspond to the predicted and true (actual) classes
- In practice, most accuracy measures are derived from this matrix

Confusion / Classification matrix

- In the two-class case (e.g., yes and no), a single prediction has four different possible outcomes

		Predicted class	
		Yes	No
Actual class	Yes	True Positive	False Negative
	No	False Positive	True Negative

- Correct classifications:
  - True Positive and True Negative
- Incorrect classifications:
  - False Positive, i.e., outcome incorrectly predicted as yes / positive
  - False Negative: i.e., outcome incorrectly predicted as no / negative

Confusion / Classification matrix

		Actual Class	
		C_1	**C_2**
Predicted Class	**C_1**	$n_{1,1}$ = number of C_1 records classified correctly	$n_{2,1}$ = number of C_2 records classified incorrectly as C_1
	C_2	$n_{1,2}$ = number of C_1 records classified incorrectly as C_2	$n_{2,2}$ = number of C_2 records classified correctly

Matrix as given in the book used in our course, i.e., "Data Mining for Business Intelligence"

Confusion / Classification matrix

		Predicted class	
		Yes	No
Actual class	Yes	True Positive	False Negative
	No	False Positive	True Negative

The prediction in the figure is a:

1. True Positive
2. False Positive
3. False Negative
4. True Negative

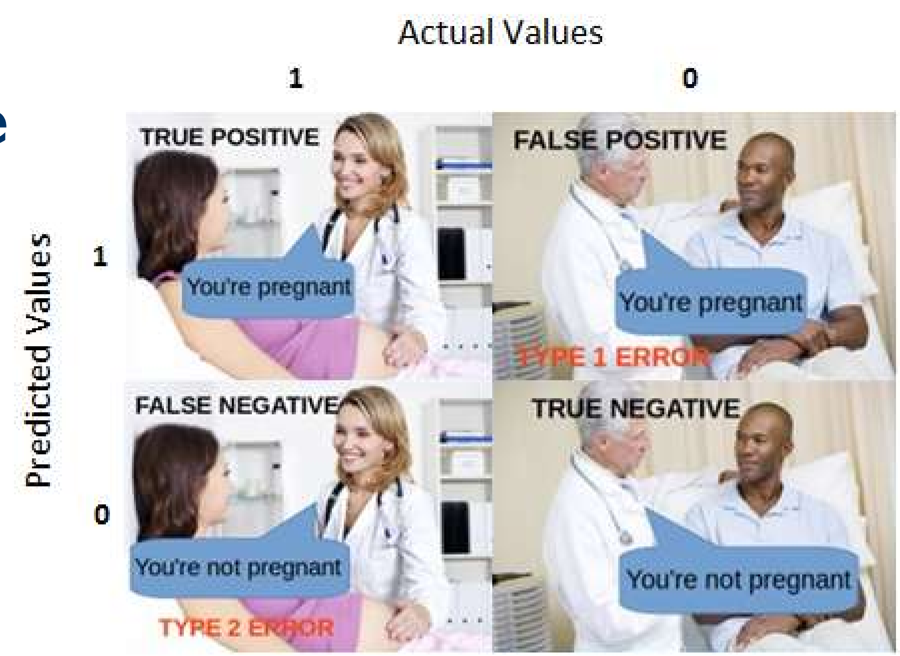
Type I and II errors

Type I Error → False Positive

- Predicted positive but that is incorrect
- Example: predicted that a man is pregnant, but actually he is not pregnant

Type II Error → False Negative

- Predicted negative but that is incorrect
- E.g., predicted a woman is not pregnant when actually she is pregnant



Overall success rate

		Predicted class	
		Yes	No
Actual class	Yes	**True Positive**	False Negative
	No	False Positive	**True Negative**

Overall success rate / Accuracy:

- Number of correct classifications divided by the total number of classifications

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \text{ or } \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{n}}$$

ROC curves (Receiver Operating Characteristic)

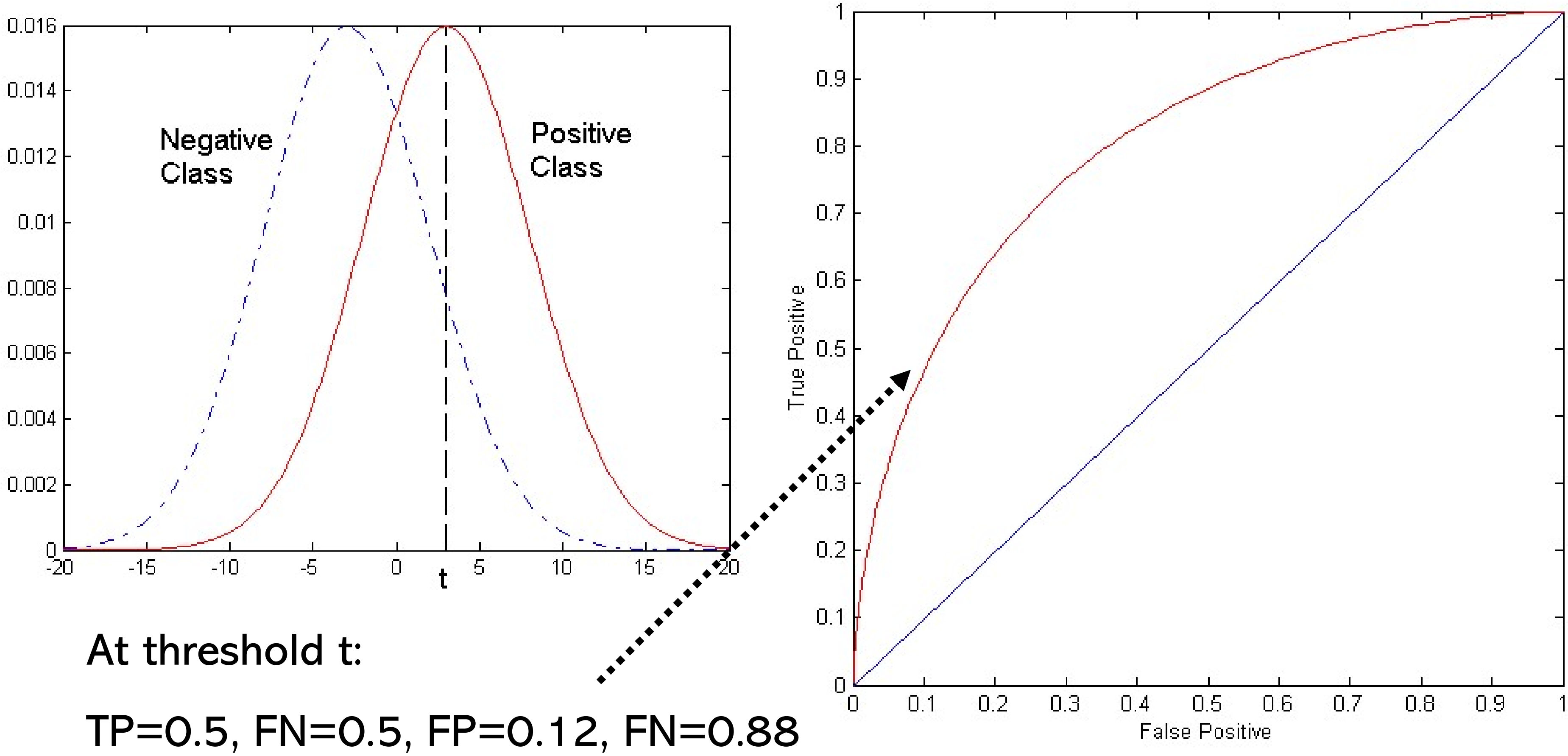
- Developed in 1950s for signal detection theory to analyze noisy signals
- Characterize the trade-off between positive hits and false alarms

- ROC curve plots TP (on the y-axis) against

  FP (on the x-axis)
- Plots the true positive rate against the false positive rate for the different possible thresholds of a diagnostic test

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:

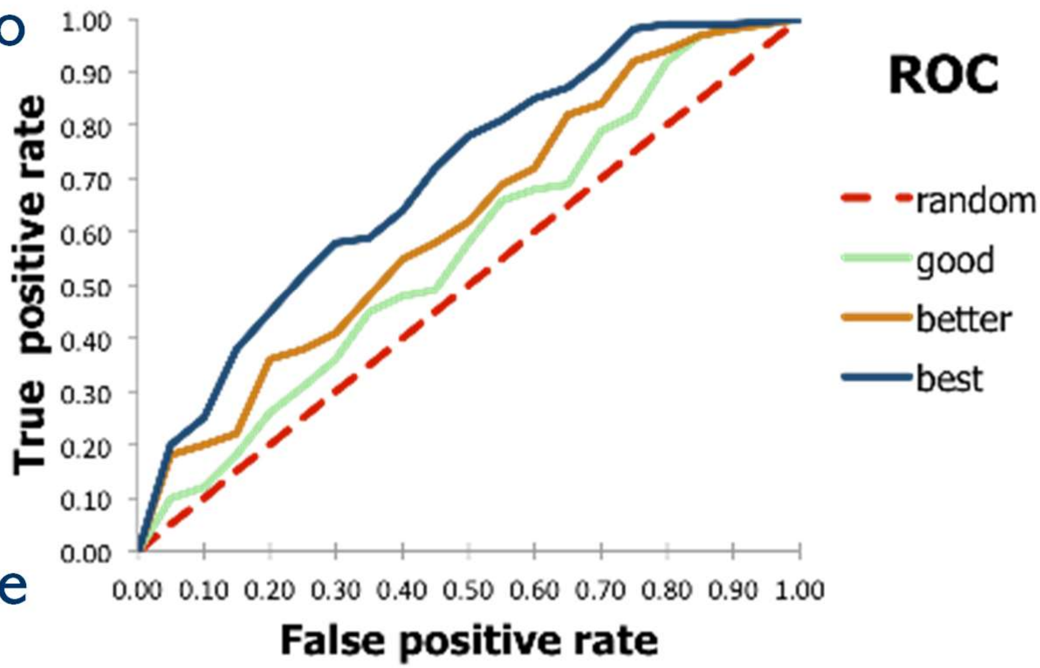
TP=0.5, FN=0.5, FP=0.12, FN=0.88

ROC Curve

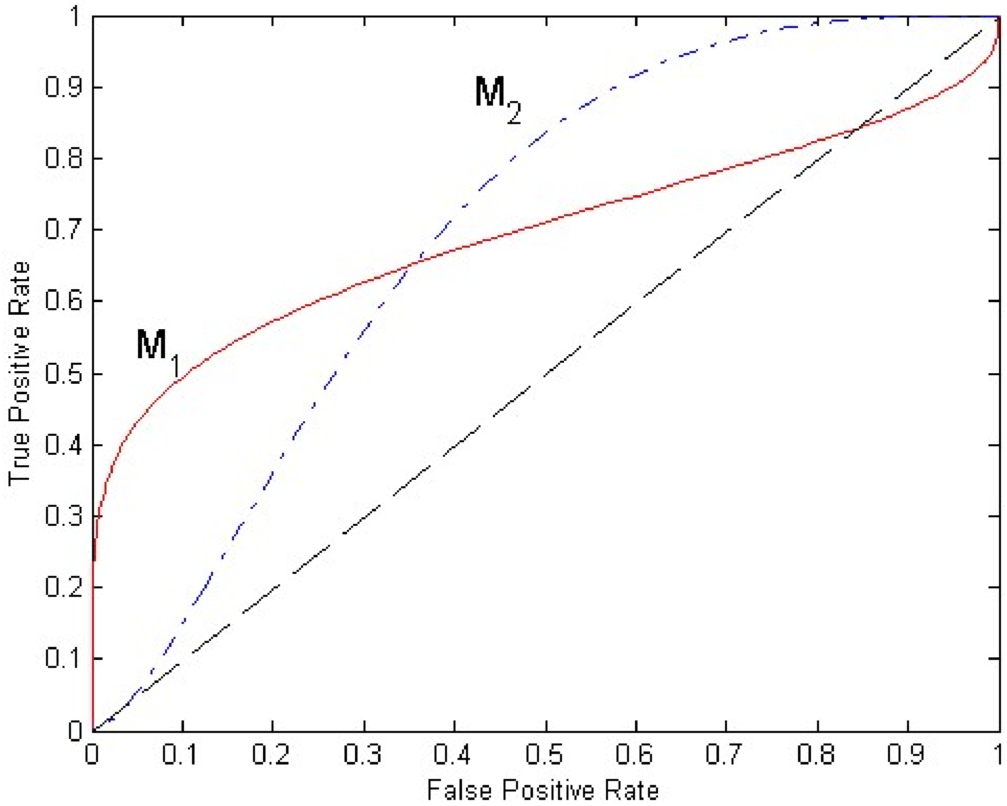
(TP,FP):

- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): 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
  - M1 is better for small FPR
  - M2 is better for large FPR

- Area Under the ROC curve
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

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 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

cost = C(Yes|Yes) x True Positive + C(No|Yes) x False Negative +
C(Yes|No) x False Positive +C(No|No) x True Negative

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. $C(Yes|Yes) = C(No|No) = p$

$$n = a + b + c + d$$

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

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

Multiclass prediction

- Two confusion matrices for a 3-class problem: actual predictor (left) vs. random predictor (right)

	Predicted Class						**Predicted Class**				
(A)		*a*	*b*	*c*	*total*	(B)		*a*	*b*	*c*	*total*
Actual class	*a*	88	10	2	*100*	**Actual Class**	*a*	60	30	10	*100*
	b	14	40	6	*60*		*b*	36	18	6	*60*
	c	18	10	12	*40*		*c*	24	12	4	*40*
	total	*120*	*60*	*20*			*total*	*120*	*60*	*20*	

- Number of successes: sum of entries in diagonal (D)
- Kappa statistic: (success rate of actual predictor - success rate of random predictor) / (1 - success rate of random predictor)
- Measures relative improvement on random predictor: 1 means perfect accuracy, 0 means we are doing no better than random

Multiclass prediction

	Predicted Class						Predicted Class				
(A)		*a*	*b*	*c*	*total*	(B)		*a*	*b*	*c*	*total*
Actual class	*a*	88	10	2	*100*	**Actual Class**	*a*	60	30	10	*100*
	b	14	40	6	*60*		*b*	36	18	6	*60*
	c	18	10	12	*40*		*c*	24	12	4	*40*
	total	*120*	*60*	*20*			*total*	*120*	*60*	*20*	

- Number of successes: sum of entries in diagonal (D)
- Kappa statistic:
  - Success rate of actual predictor – success rate of random predictor) / (1 – success rate of random predictor)
- Example:
  - Success rate of actual predictor = (88+40+12)/200 = 140/200 = 0.7
  - Success rate of random predictor = (60+18+4)/200 = 82/200 = 0.41
  - Kappa statistics = (0.7–0.41)/(1–0.41) = 0.492

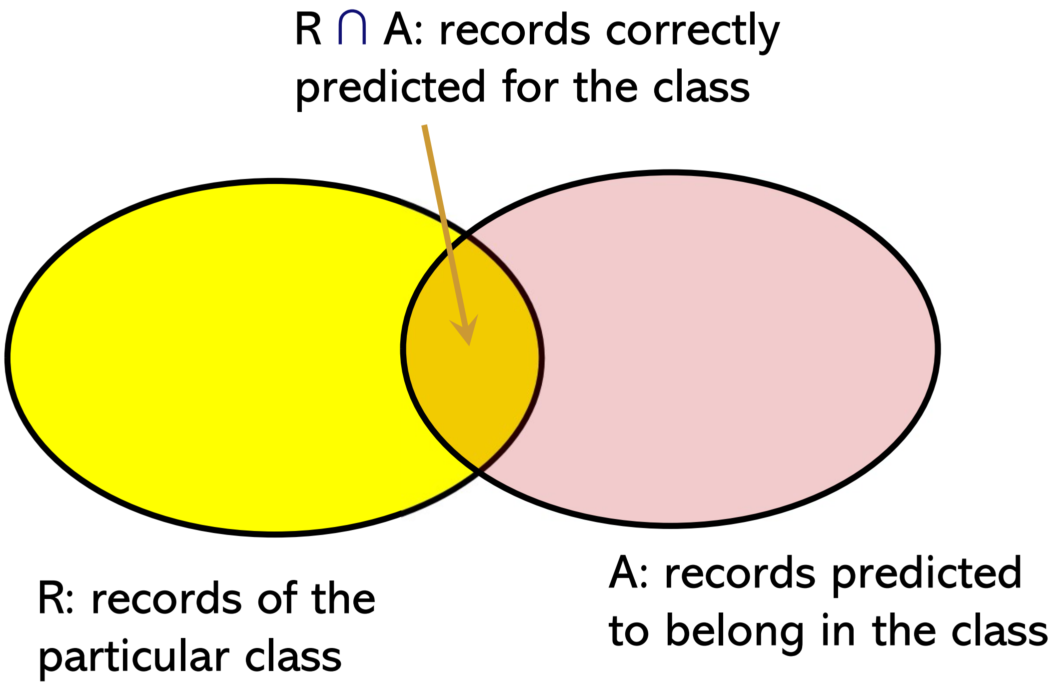
Precision and Recall

Precision and Recall

R: documents of a particular class from the testing set

A: prediction set (answers) for the particular class, generated by a model

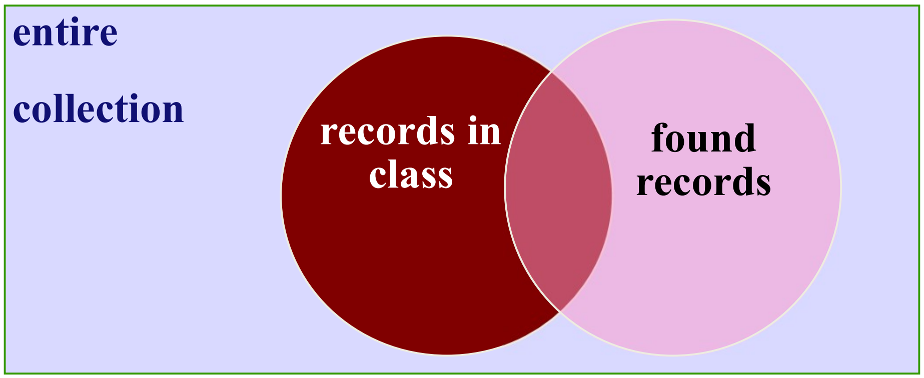
R ∩ A: the intersection of the sets R and A

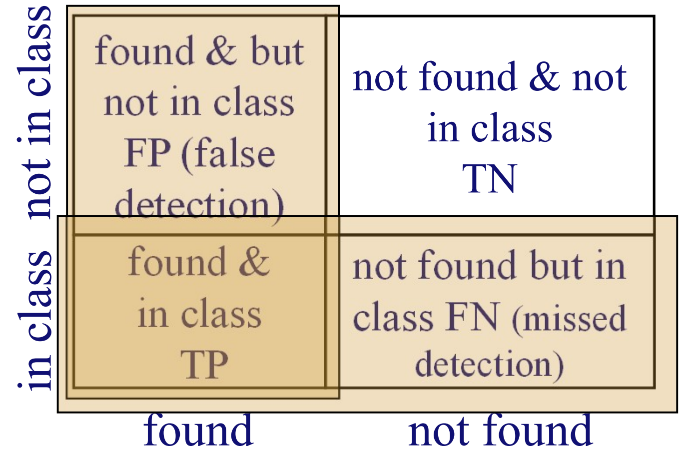


$$Recall = \frac{|R \cap A|}{|R|}$$

$$Precision = \frac{|R \cap A|}{|A|}$$

Precision and Recall





$$\text{recall} = \frac{\textit{number of records found that belong to the particular class}}{\textit{number of records in class}}$$

The ability of the model to find ***all*** of the items of the class

$$\text{precision} = \frac{\textit{number of records found that belong to the particular class}}{\textit{number of records found for the particular class}}$$

The ability of the model to correctly detect class items

Determining Recall is Difficult

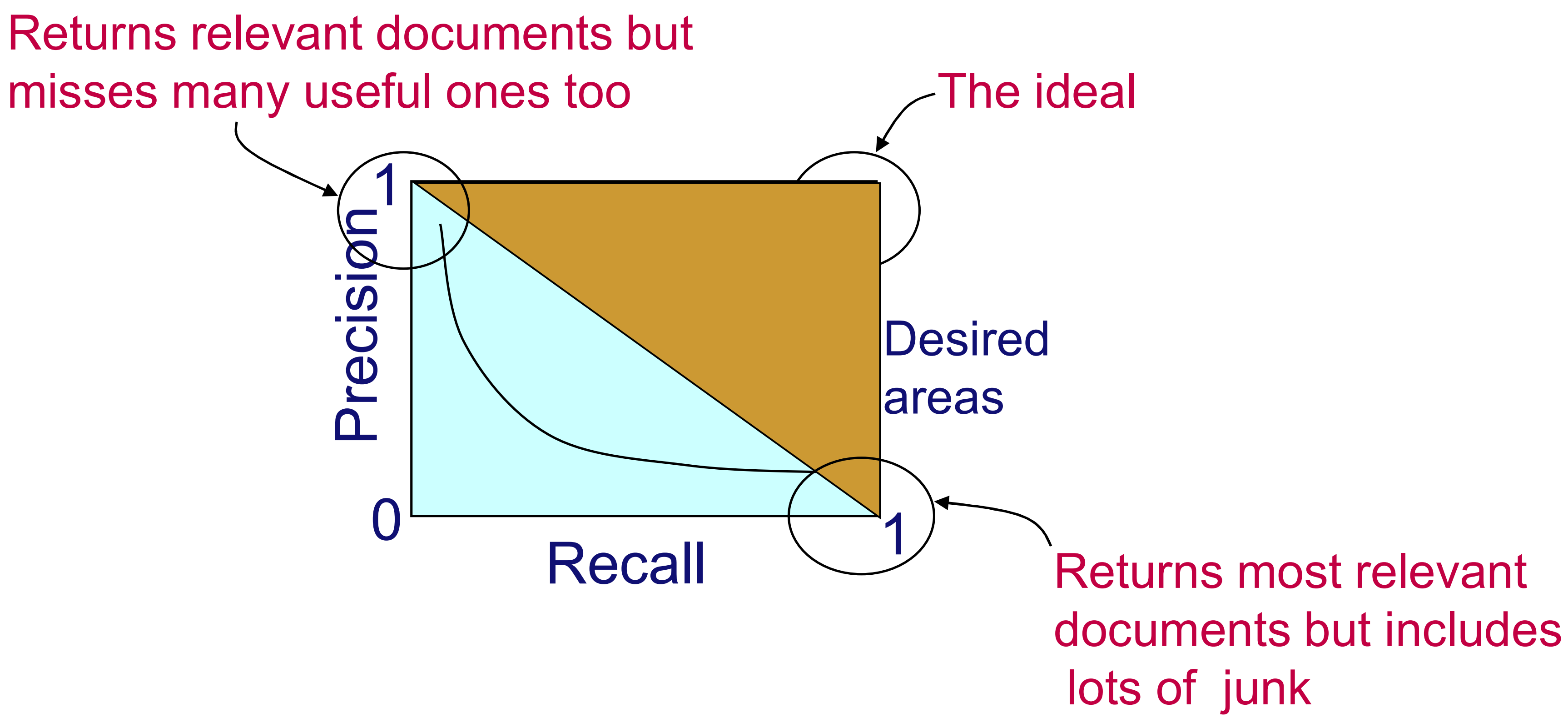
- Total number of items / records that belong to a particular class is sometimes not available

Solutions:

- Sample across the dataset and perform relevance judgment on these items
- Apply different models to the same dataset and then use the aggregate of relevant items as the total relevant set

Evaluating ranked results: R&P tradeoff

- The system can return any number of results
- By taking various numbers of the top returned documents (levels of recall), the evaluator can produce a *precision-recall curve*



F-Measure (F1-measure)

- One measure of performance that takes into account both recall and precision
- Harmonic mean of recall and precision:

$$F = \frac{2PR}{P+R} = \frac{2}{\frac{1}{R}+\frac{1}{P}}$$

Question for you:

Consider two systems A and B, their IR routines are identical except that A does stemming as preprocessing.

- How precision and recall of A and B relate to each other?

A. B has higher precision and lower recall than A.

B. A and B have about the same recall.

C. A and B have about the same precision and about the same recall.

D. B has lower precision and higher recall than A.

E. A and B have about the same precision.

F. B has higher precision and higher recall than A.

G. B has lower precision and lower recall than A.

H. Impossible to say

Training and Validation Performance

- Errors that are based on the training set tell us about model fit
- Errors that are based on the validation set measure the model's ability to predict new data
  - Prediction errors that measure predictive performance

What is the expected relationship between training errors and validation errors?

a) Training errors are less than the validation errors

b) Validation errors are less than the training errors

c) Cannot predict the relationship

d) Exactly identical

Confusion / Classification matrix

		Predicted class	
		Yes	No
Actual class	Yes	True Positive	False Negative
	No	False Positive	True Negative

The prediction in the figure is a:

1. True Positive
2. False Positive
3. False Negative
4. True Negative