CSMDM21 - Data Analytics and Mining

Classification Model Evaluation

Module convenor
Dr. Carmen Lam
carmen.lam@reading.ac.uk
Department of Computer Science

Lecture notes and videos created by
Prof. Giuseppe Di Fatta

University of Reading

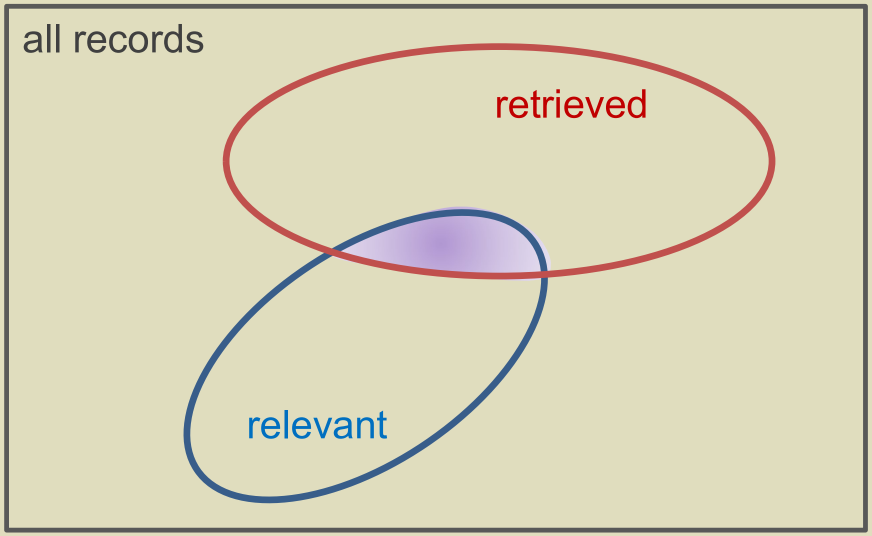
Model Evaluation

- 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?

Evaluation of Predictive Tasks

- For supervised classification we have a variety of measures to evaluate how good our model is
  - Classification task
    - **Accuracy**: the fraction of classifications that are correct
    - **Error rate** = 1 - accuracy
  - Information Retrieval task
    - **Recall**: proportion of relevant material actually retrieved
    - **Precision**: proportion of retrieved material actually relevant

Information Retrieval Systems



$$\text{Precision} = \frac{|\text{Relevant Retrieved}|}{|\text{Retrieved}|}$$

$$\text{Recall} = \frac{|\text{Relevant Retrieved}|}{|\text{Relevant in Collection}|}$$

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)

Accuracy

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

- Most widely-used metric:

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

$$\text{Error rate} = \frac{b + c}{a + b + c + d} = \frac{FN + FP}{TP + TN + FP + FN} = 1 - Accuracy$$

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

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

Measures Beyond Accuracy

$$\text{Precision} = \frac{|\text{Relevant Retrieved}|}{|\text{Retrieved}|} = \frac{a}{a+c}$$

$$\text{Recall} = \frac{|\text{Relevant Retrieved}|}{|\text{Relevant in Collection}|} = \frac{a}{a+b}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

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

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

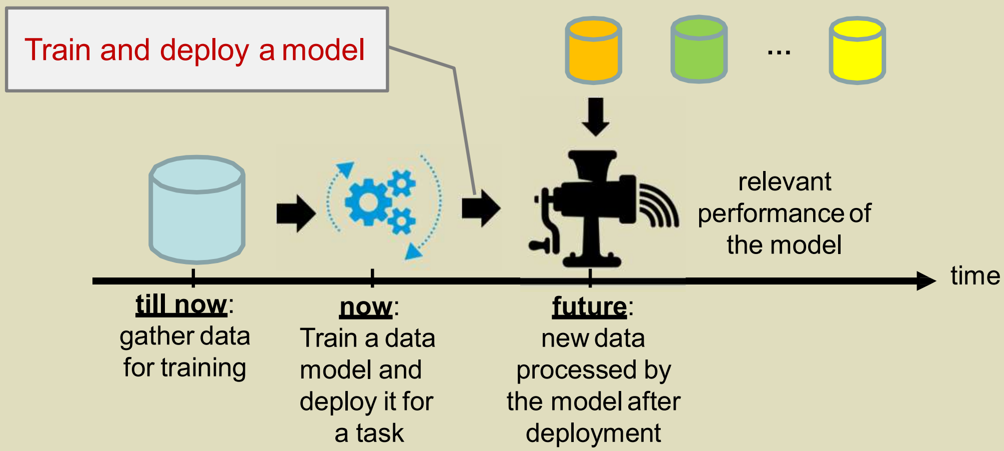
- Precision is biased towards C(Yes|Yes) & C(Yes|No)
- Recall is biased towards C(Yes|Yes) & C(No|Yes)
- F-measure is biased towards all except C(No|No)
  - The F-measure is also referred to as F-score or F_1 score. It is the harmonic mean of the precision and recall. The max value is 1 (perfect precision and recall) and the min value is 0 (precision or recall is zero).
- Generalised F-measure: recall is β times more important than precision.

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall}$$

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?

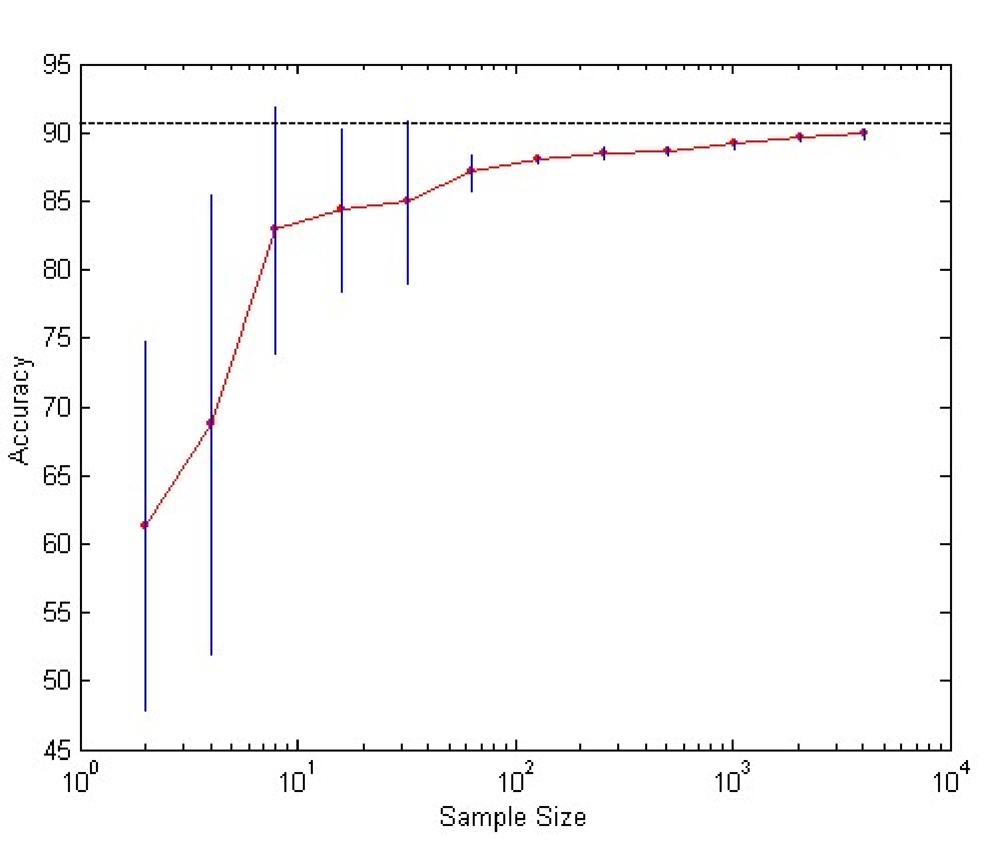
Methods for Performance Evaluation



How to obtain a reliable estimate of "future" performance?

- Performance of a model may depend on other factors besides the learning algorithm:
  - Class distribution
  - Cost of misclassification
  - Size of training dataset
  - Data distributions (present vs future)

Learning Curve: Accuracy vs Sample Size

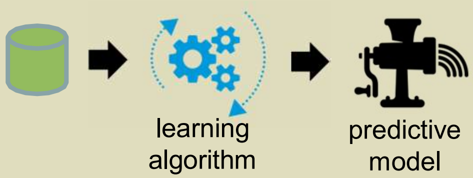


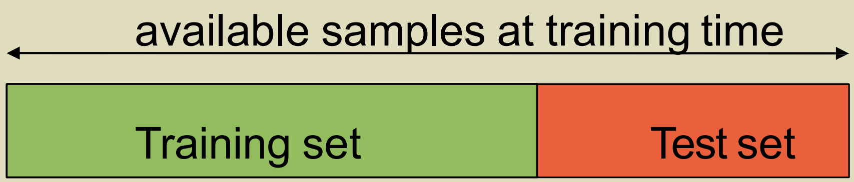
- The learning curve shows how accuracy changes with varying sample size
  - Requires a sampling method for creating a learning curve
- Effect of small sample size:
  1. Bias in the estimate
  2. Variance of the estimate

The Holdout Method

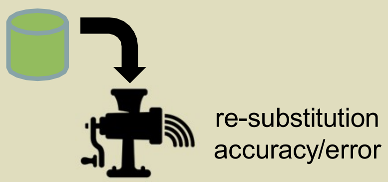
- Split the available data into two disjoint partitions
  - Training set: used to train the classifier
  - Test set: <u>hold out some data</u> to estimate the error rate of the trained classifier

<u>Training the model:</u>

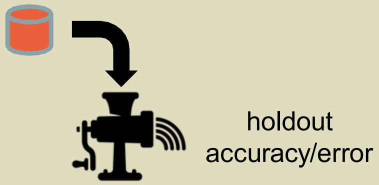


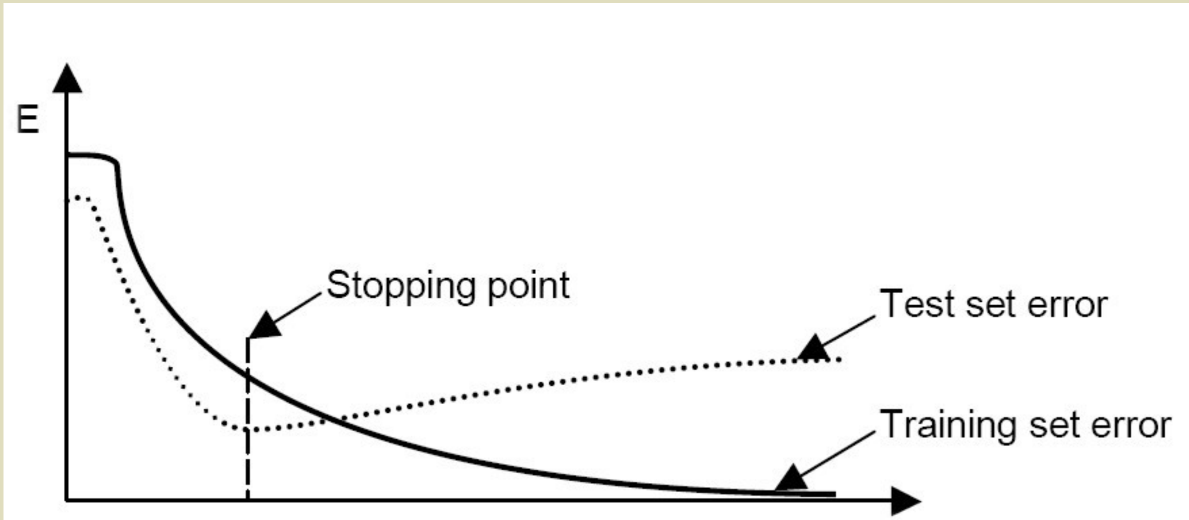


<u>Testing the model on the training data:</u>

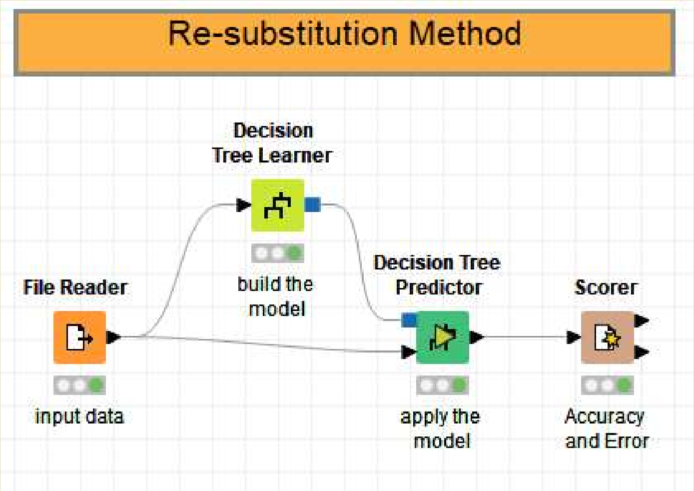


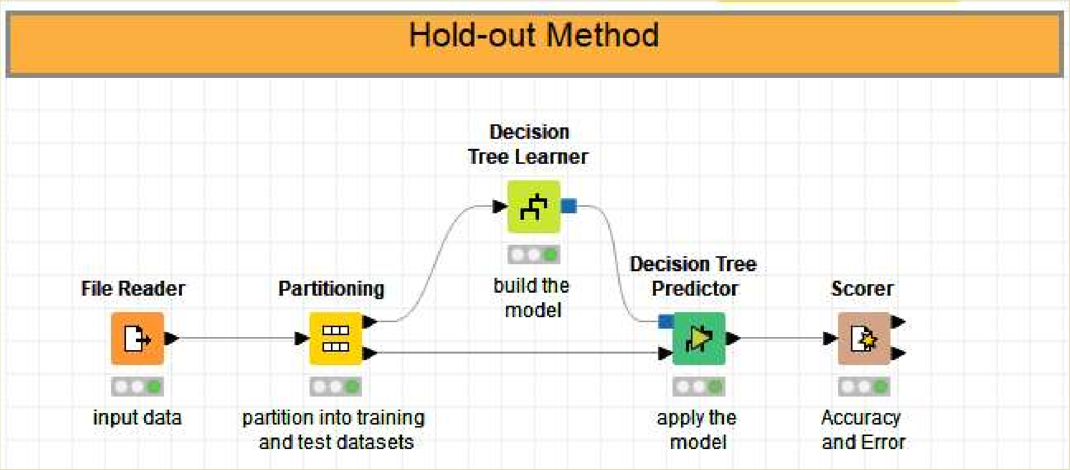
<u>Testing the model on the test data:</u>





Re-substitution and Holdout Methods in KNIME





The Holdout Method

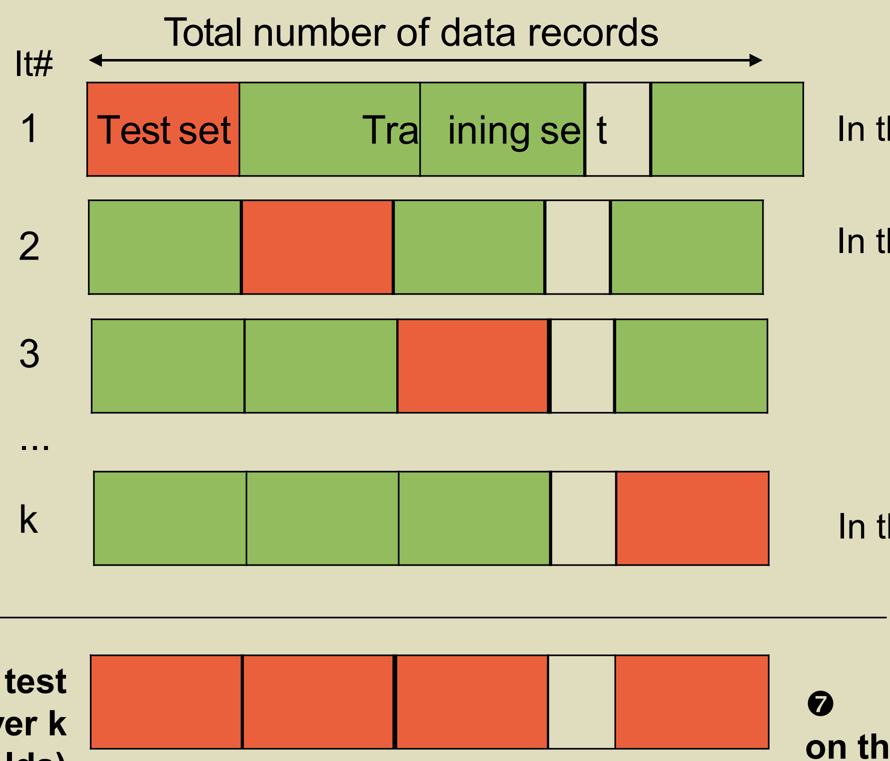
- Holdout method
  - Original data is partitioned into two disjoint sets
  - E.g. 2/3 for training and 1/3 for testing, or 50%-50%

- Issues
  - Less data available for training
  - Bias on the training set
  - Training set and test set are not independent (e.g. there may be unbalanced classes in both)

Methods of Estimation

- Holdout
  - Original data is partitioned into two disjoint sets
- Random subsampling
  - Repeated holdout
    - Still similar issues to simple holdout
    - No control over the selected samples (samples may be chosen multiple times or never)
- 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, where n is the number of data samples
- Bootstrap
  - Sampling with replacement
  - .632 bootstrap

k-fold Cross-validation (xval)



In the 1st fold, hold the 1st partition out

In the 2nd fold, hold the 2nd partition out

In the last fold, hold the last partition out

accumulated test results over k iterations (folds)

❼ accuracy and error rate are computed on the results over all available data.

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 TP (on the y-axis) against FP (on the x-axis)
  - ➢ **True Positive rate vs False Positive rate**

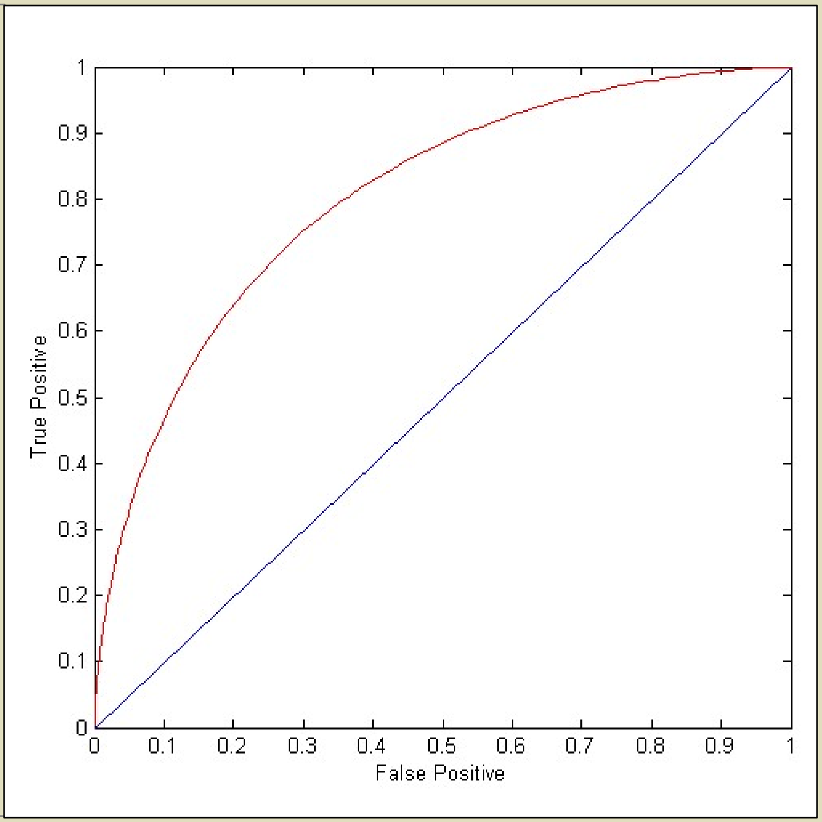
- Performance of each classifier represented as a point on the ROC curve
  - changing some critical parameter of the algorithm (e.g., sample distribution, cost matrix, or any hyperparameters) determines a set of points (TP,FP).

ROC Curve of a Classifier

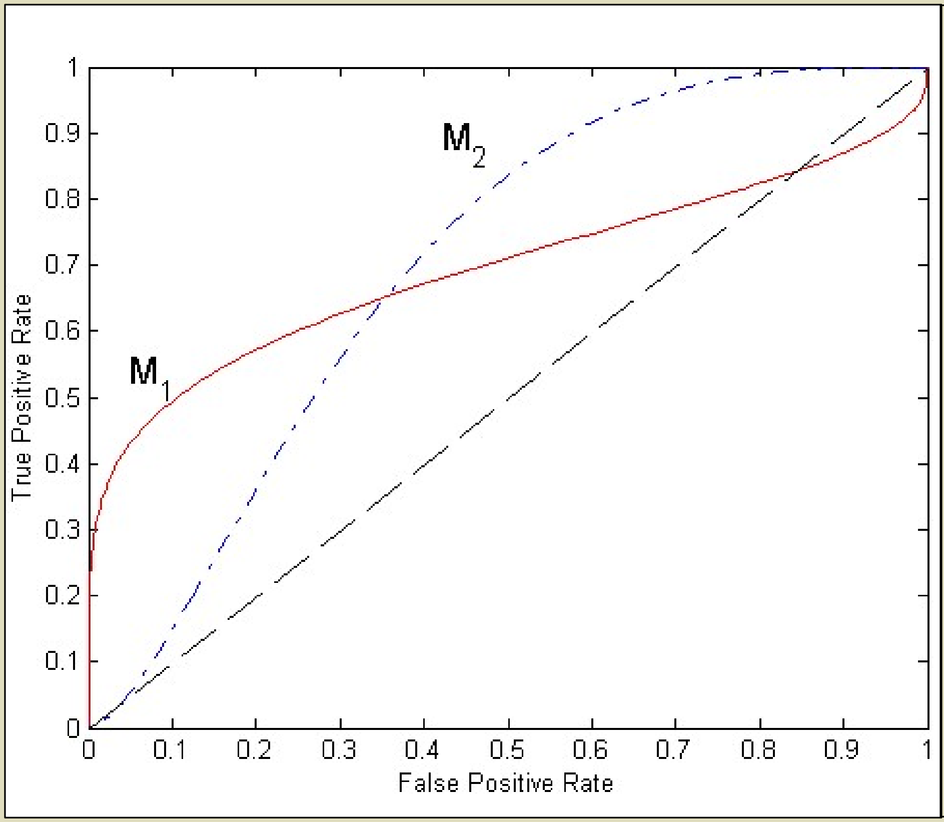
(TP,FP)

- (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



M_1 vs M_2

- 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 (AUC)
  - Ideal:
    - Area = 1
  - Random guess:
    - Area = 0.5

Next:

- P04: practical on Classification in KNIME

Next week:

- No teaching (for you to catch up on lectures and exercises)

Week 7:

- Advanced KNIME