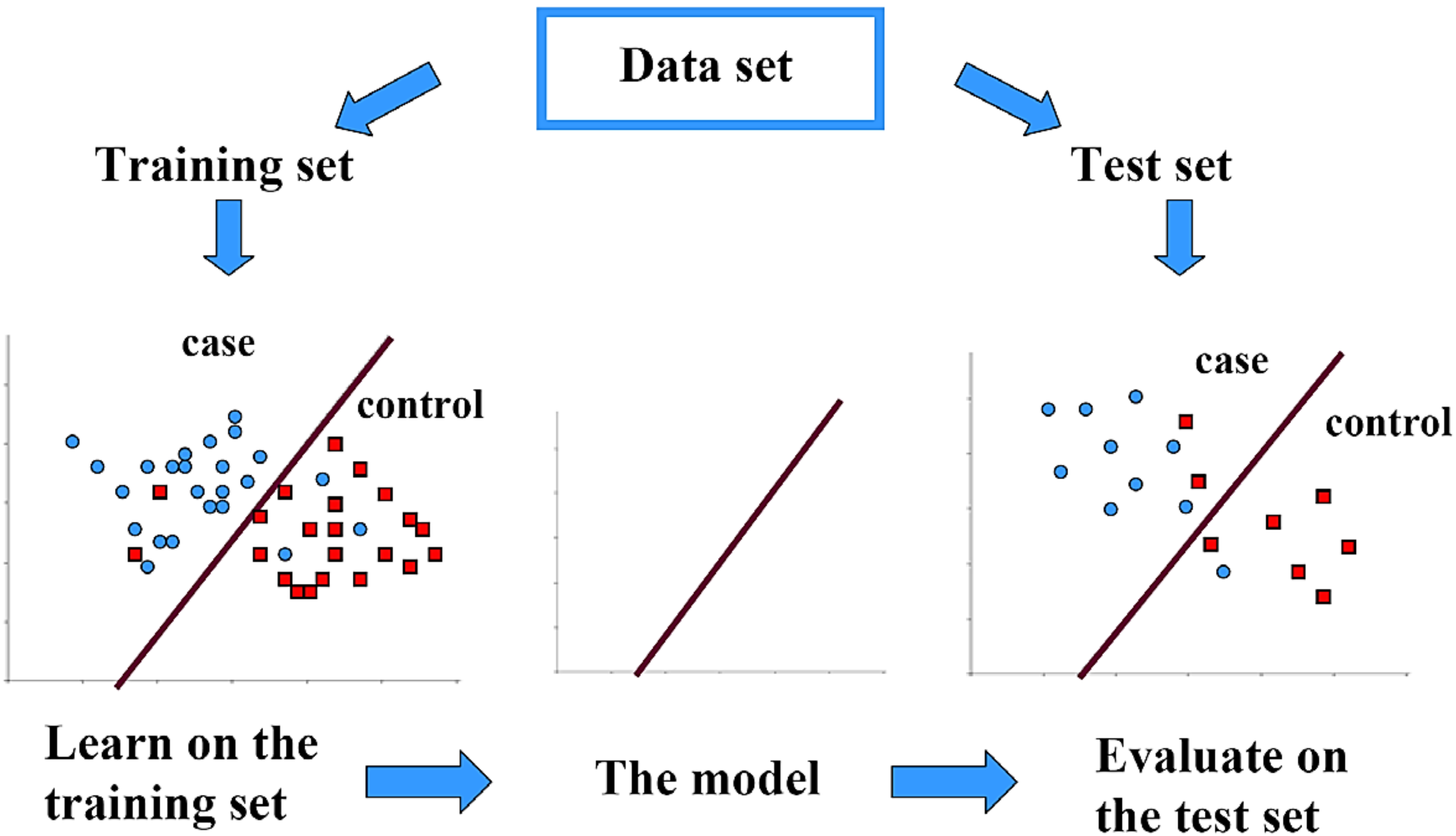
INTRODUCTION TO MACHINE LEARNING

EVALUATION

* Some contents are adapted from Dr. Hung Huang and Dr. Chengkai Li at UT Arlington

Mingon Kang, PhD
Department of Computer Science @ UNLV

Evaluation for Classification



Evaluation Metrics

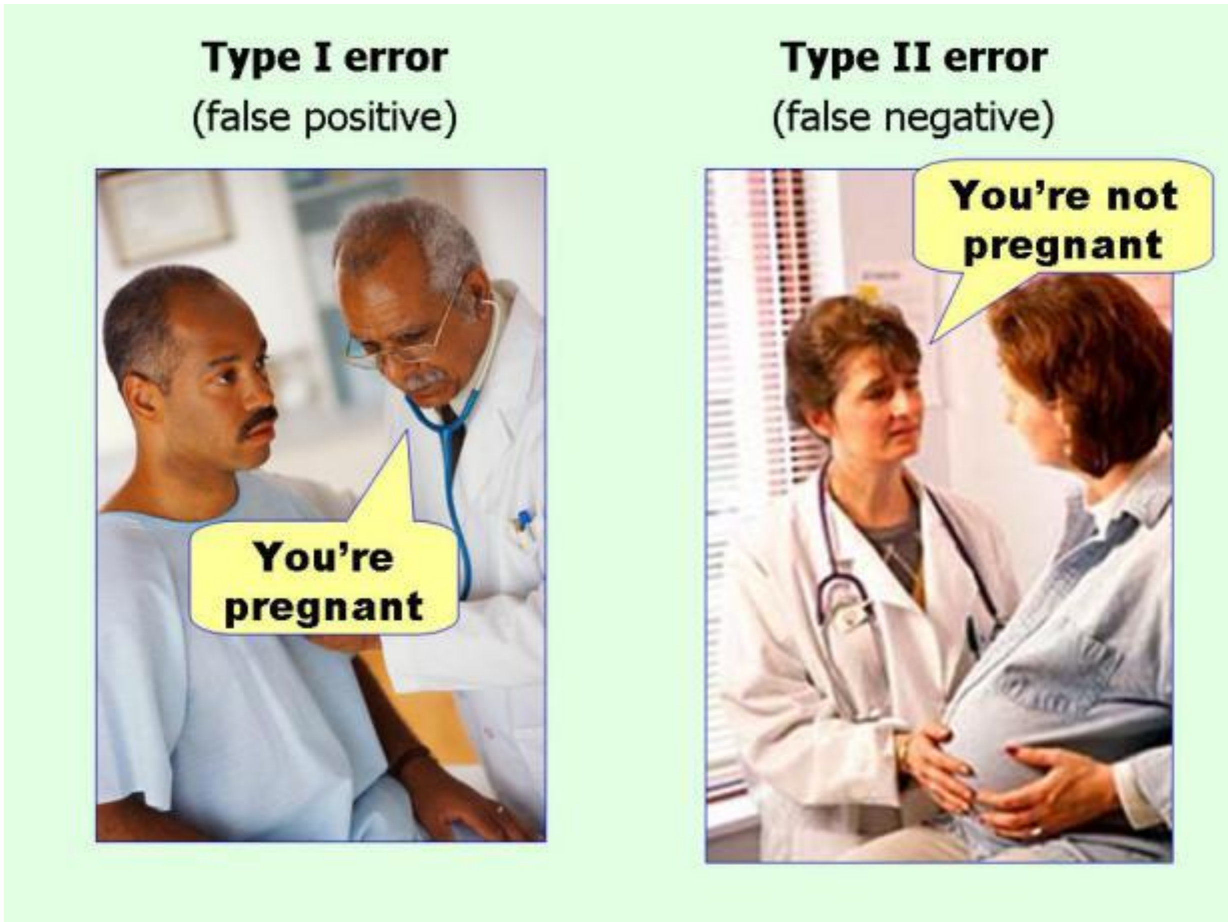
- Confusion Matrix: shows performance of an algorithm, especially predictive capability.
  - rather than how fast it takes to classify, build models, or scalability.

	Predicted Class		
Actual Class		Class = YES	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Evaluation Metrics

		Predicted condition			
	Total population	Predicted Condition positive	Predicted Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
True condition	condition positive	**True positive**	**False Negative** (Type II error)	True positive rate (TPR), Sensitivity, Recall $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$
	condition negative	**False Positive** (Type I error)	**True negative**	False positive rate (FPR), Fall-out $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	True negative rate (TNR), Specificity (SPC) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$
	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	

Type I and II error



Evaluation Metrics

- Sensitivity or True Positive Rate (TPR)
  - TP/(TP+FN)
- Specificity or True Negative Rate (TNR)
  - TN/(FP+TN)
- Precision or Positive Predictive Value (PPV)
  - TP/(TP+FP)
- Negative Predictive Value (NPV)
  - TN/(TN+FN)
- Accuracy
  - (TP+TN)/(TP+FP+TN+FN)

Limitation of Accuracy

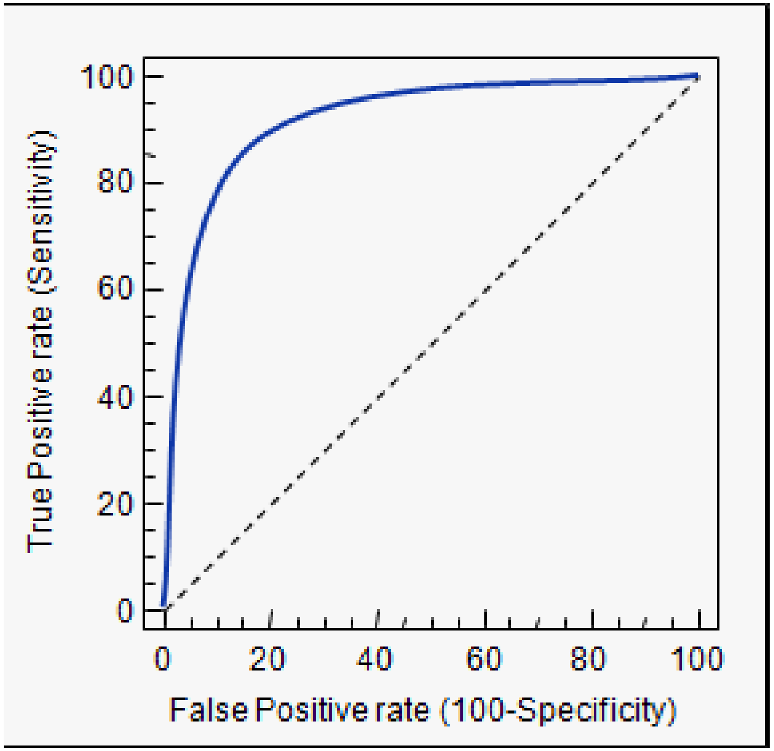
- Consider a binary classification problem
  - Number of Class 0 examples = 9990
  - Number of Class 1 examples = 10
  - If predict all as 0, accuracy is 9990/10000=99.9%

- Precision
- Recall
- Weighted Accuracy $= \dfrac{w_{TP}TP + w_{TN}TN}{w_{TP}TP + w_{FP}FP + w_{TN}TN + w_{FN}FN}$

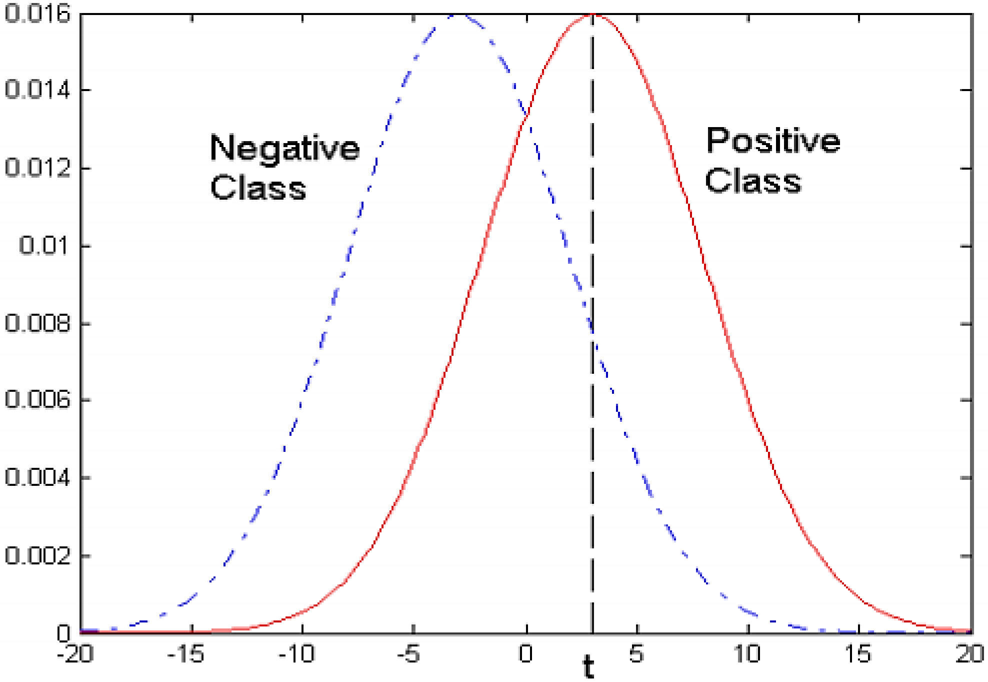
ROC curve

- Receiver Operating Characteristic
  - Graphical approach for displaying the tradeoff between true positive rate(TPR) and false positive rate (FPR) of a classifier
    - TPR = positives correctly classified/total positives
    - FPR = negatives incorrectly classified/total negatives
  - TPR on y-axis and FPR on x-axis

ROC curve


True Positive rate (Sensitivity)
100
80
60
40
20
0
0
20
40
60
80
100
False Positive rate (100-Specificity)

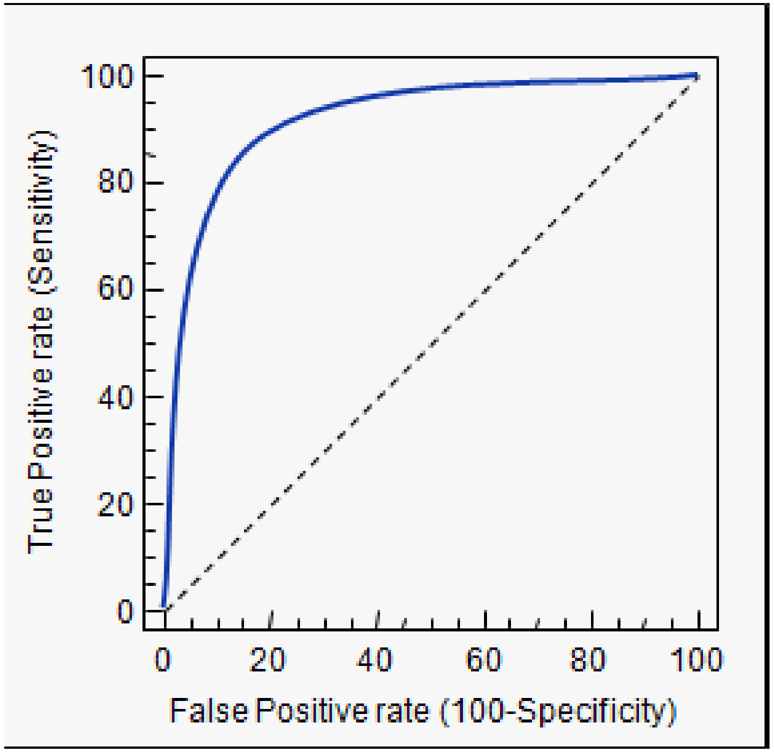
ROC curve



0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
-20
-15
-10
-5
0
5
10
15
20
t
Negative Class
Positive Class

ROC curve

- Points of interests (TP, FP)
  - (0, 0): everything is negative
  - (1, 1): everything is positive
  - (1, 0): perfect (ideal)
- Diagonal line
  - Random guessing (50%)
- Area Under Curve (AUC)
  - Measurement how good the model on the average
  - Good to compare with other methods



Evaluation

- Model Selection
  - How to evaluate the performance of a model?
  - How to obtain reliable estimates?
- Performance estimation
  - How to compare the relative performance with competing models?

Motivation

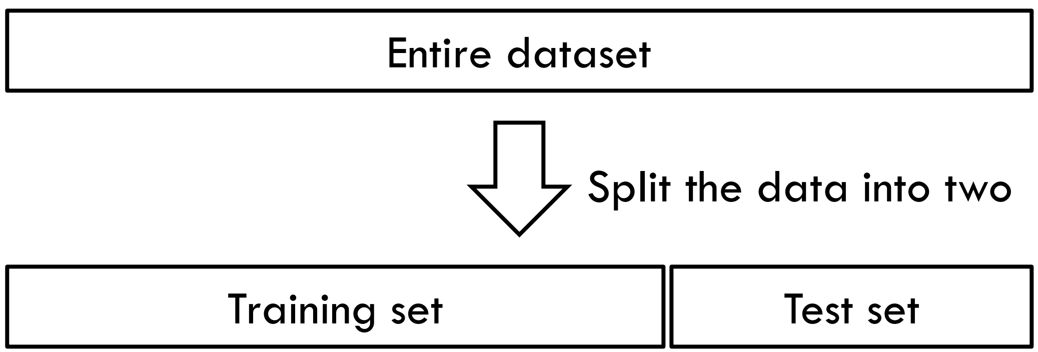
- We often have a finite set of data
  - If using the entire training data for the best model,
    - The model normally overfits the training data, where it often gives almost 100% correct classification results on training data
- Better to split the training data into disjoint subsets
- Note that test data is not used in any way to create the classifier → Cheating!

Methods of Validation

- Holdout
  - Use 2/3 for training and 1/3 for testing
- Cross-validation
  - Random subsampling
  - K-Fold Cross-validation
  - Leave-one-out
- Stratified cross-validation
  - Stratified 10-fold cross-validation is often the best
- Bootstrapping
  - Sampling with replacement
  - Oversampling vs undersampling

Holdout

- Split dataset into two groups for training and test
  - Training dataset: used to train the model
  - Test dataset: use to estimate the error rate of the model



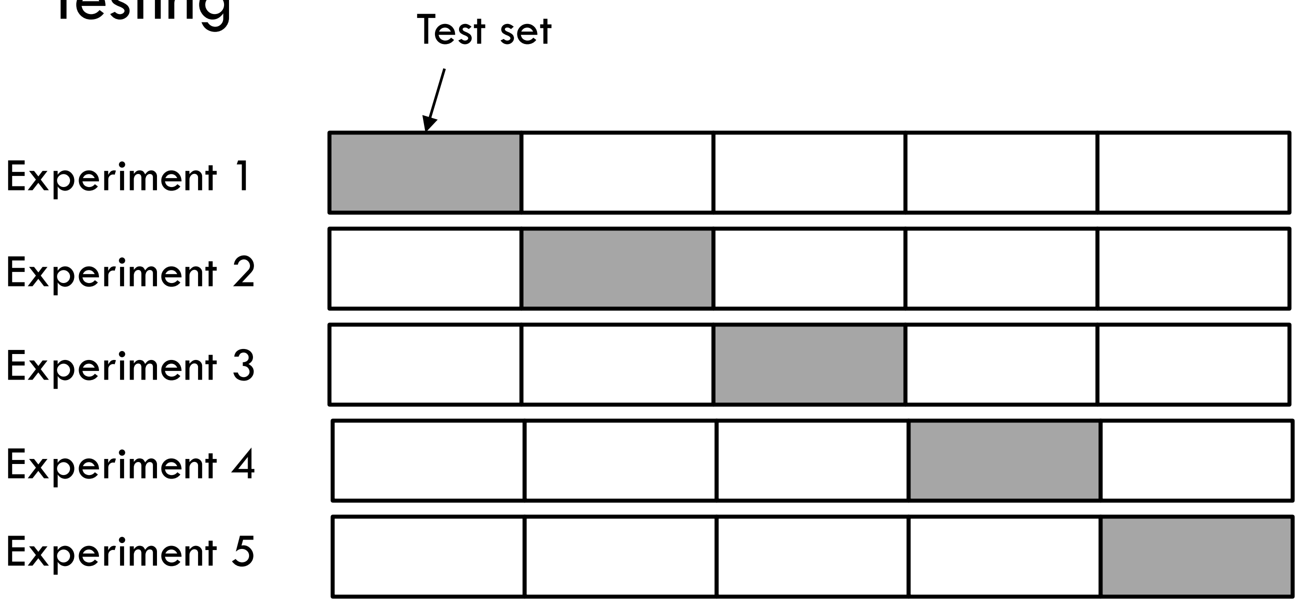
- Drawback
  - When “unfortunate split” happens, the holdout estimate of error rate will be misleading

Random Subsampling

- Split the data set into two groups
  - Randomly selects a number of samples without replacement
    - Usually, one third for testing, the rest for training

K-Fold Cross-validation

- K-fold Partition
  - Partition K equal sized sub groups
  - Use K-1 groups for training and the remaining one for testing



K-fold cross-validation

- Suppose that E_i is the performance in the *i*-th experiment
- The average error rate is

$$E = \frac{1}{K}\sum_{i=1}^{k} E_i$$

Leave-one-out cross-validation

- Use N-1 samples for training and the remaining sample for testing (i.e., there is only one sample for testing)
- The average error rate is

$$E = \frac{1}{N}\sum_{i=1}^{N} E_i$$

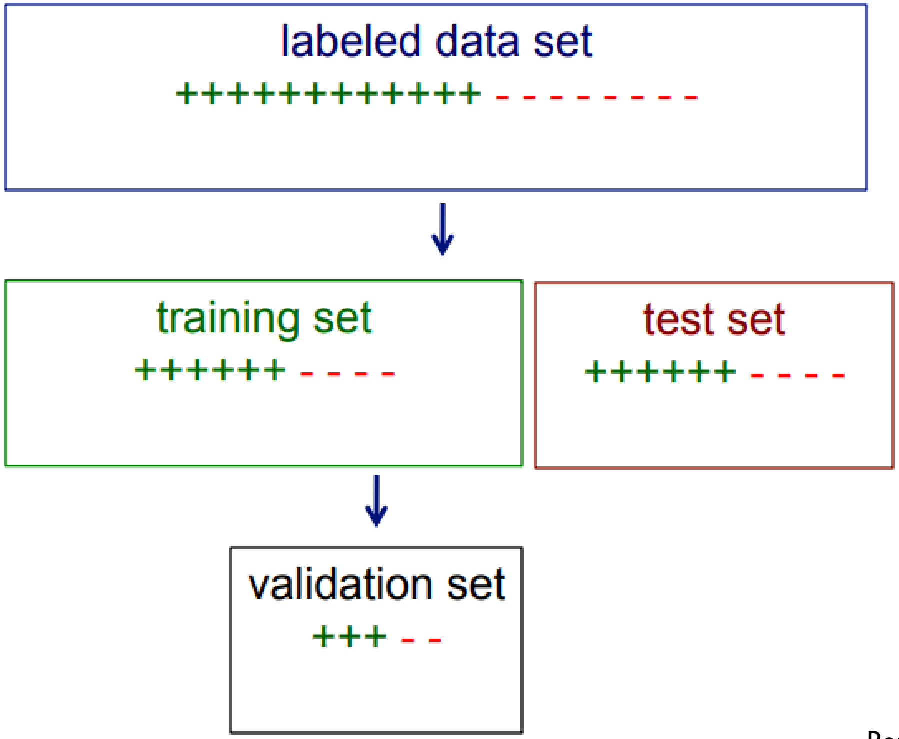
where N is the total sample number.

How many folds?

- If a large number of folds
  - Bias to the true estimator will be small.
  - The estimator will be accurate
  - Computationally expensive
- If a small number of folds
  - Cheap computational time for experiments
  - Variance of the estimator will be small
  - Bias will be large
- 5 or 10-Fold CV is a common choice for K-fold CV

Stratified cross-validation

- When randomly selecting training or test sets, ensure that class proportions are maintained in each selected set.

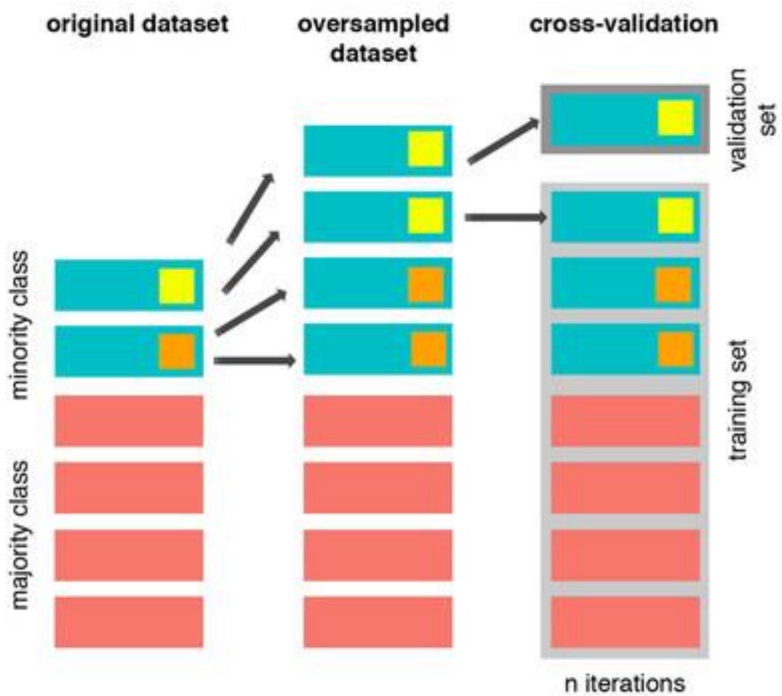


1. Stratify instances by class
2. Randomly select instances from each class proportionally

Ref: http://pages.cs.wisc.edu/~dpage/cs760/evaluating.pdf

Bootstrapping

- Oversampling
  - Amplifying the minor class samples so that the classes are equally distributed



Bootstrapping

- Undersampling
  - Consider less numbers of samples in the major class so that the classes are equally distributed

Cross-validation with normalization

- Cross-validation with normalization
  - The model is optimized to the normalized data rather than the original data
  - How to evaluate via CV with normalization (e.g., z-score normalization)?
    - Normalize the training data (obtain mean and std)
    - Normalize the validation or test data with the mean and std obtained from the training data
      - Otherwise, the test data are not independent from the training data. Weak cheating.