# Artificial Intelligence for Medicine II

Spring 2025

## Lecture 4: Supervised Learning Model Evaluation and Evaluation Metrics

(Many slides adapted from Bing Liu, Han, Kamber & Pei; Tan, Steinbach, Kumar and the web)

# Model Evaluation

- Regression Error Measures
- Model Evaluation
  - Metrics for Performance Evaluation
- How to evaluate the performance of a model
  - Methods for Performance Evaluation

# Regreassion Error Measures

- Measure predictor accuracy: measure how far off the predicted value is from the actual known value
- **Loss function**: measures the error betw. $y_i$ and the predicted value $y_i'$
  - Absolute error: $| y_i - y_i' |$
  - Squared error: $(y_i - y_i')^2$
- Test error (generalization error): the average loss over the test set
  - Mean absolute error: $\frac{\sum_{i=1}^{d} | y_i - y_i' |}{d}$ Mean squared error: $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$
  - Relative absolute error: $\frac{\sum_{i=1}^{d} | y_i - y_i' |}{\sum_{i=1}^{d} | y_i - \bar{y} |}$ Relative squared error: $\frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2}$

The mean squared-error exaggerates the presence of outliers

Popularly use (square) root mean-square error, similarly, root relative squared error

# Metrics for Performance Evaluation: Classification Accuracy

- Classification accuracy is usually calculated by determining the percentage of records(tuples) placed in the correct class.
- Given a specific class, $C_j$, and a database tuple, $t_i$,
- The tuple, $t_i$, may or may not be assigned to that class while its actual membership may or may not be in that class
- This can be described in the following ways

| Class | Prediction | Actual |
|---|---|---|
| True positive (TP) | $t_i$ in $C_j$ | $t_i$ in $C_j$ |
| False positive (FP) | $t_i$ in $C_j$ | $t_i$ not in $C_j$ |
| True negative (TN) | $t_i$ not in $C_j$ | $t_i$ not in $C_j$ |
| False negative (FN) | $t_i$ not in $C_j$ | $t_i$ in $C_j$ |

# Confusion Matrix

- A confusion matrix (Kohavi and Provost, 1998) contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix.
- Given m classes, a confusion matrix is an *m x m* matrix where $c_{i,j}$ indicates the number of tuples from D that were assigned to $C_j$ but the correct class $C_i$

# Example for Confusion Matrix

| Name | Gender | Height | Actual | Assigned |
|---|---|---|---|---|
| Kristina | F | 1.6m | Short | Medium |
| Jim | M | 2m | Tall | Medium |
| Maggie | F | 1.9m | Medium | Tall |
| Martha | F | 1.88m | Medium | Tall |
| Stephanie | F | 1.7m | Short | Medium |
| Bob | M | 1.85m | Medium | Medium |
| Kathy | F | 1.6m | Short | Medium |
| Dave | M | 1.7m | Short | Medium |
| Worth | M | 2.2m | Tall | Tall |
| Steven | M | 2.1m | Tall | Tall |
| Debbie | F | 1.8m | Medium | Medium |
| Todd | M | 1.95m | Medium | Medium |
| Kim | F | 1.9m | Medium | Tall |
| Amy | F | 1.8m | Medium | Medium |
| Wynette | F | 1.75m | Medium | Medium |

| Actual Membership | Assignment | | |
|---|---|---|---|
| | Short | Medium | Tall |
| Short | 0 | 4 | 0 |
| Medium | 0 | 5 | 3 |
| Tall | 0 | 1 | 2 |

# Example for Confusion Matrix

Predicted

| Actual | | $C_1$ | $C_2$ |
|---|---|---|---|
| | $C_1$ | True positive | False negative |
| | $C_2$ | False positive | True negative |

| classes | buy_computer = yes | buy_computer = no | total | recognition(%) |
|---|---|---|---|---|
| buy_computer = yes | 6954 | 46 | 7000 | 99.34 |
| buy_computer = no | 412 | 2588 | 3000 | 86.27 |
| total | 7366 | 2634 | 10000 | 95.52 |

- Accuracy of a classifier M, acc(M): percentage of test set tuples that are correctly classified by the model M
  - Error rate (misclassification rate) of M = 1 – acc(M)
  - Given *m* classes, $CM_{i,j}$, an entry in a **confusion matrix**, indicates # of tuples in class *i* that are labeled by the classifier as class *j*

# Evaluating classification methods

- **Predictive accuracy**

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}}$$

- Efficiency
  - time to construct the model
  - time to use the model
- Robustness: handling noise and missing values
- Scalability: efficiency in disk-resident databases
- Interpretability:
  - understandable and insight provided by the model
- Compactness of the model: size of the tree, or the number of rules.

# Classification measures

- Accuracy is only one measure (error = 1-accuracy).
- **Accuracy is not suitable in some applications**.
- In text mining, we may only be interested in the documents of a particular topic, which are only a small portion of a big document collection.
- In classification involving skewed or highly imbalanced data, e.g., network intrusion and financial fraud detections, we are interested only in the minority class.
  - High accuracy does not mean any intrusion is detected.
  - E.g., 1% intrusion. Achieve 99% accuracy by doing nothing.
- The class of interest is commonly called the **positive class**, and the rest **negative classes.**

# Precision and recall measures

- Used in information retrieval and text classification.
- We use a confusion matrix to introduce them.

| | Classified Positive | Classified Negative |
|---|---|---|
| Actual Positive | TP | FN |
| Actual Negative | FP | TN |

where

*TP*: the number of correct classifications of the positive examples (**true positive**),

*FN*: the number of incorrect classifications of positive examples (**false negative**),

*FP*: the number of incorrect classifications of negative examples (**false positive**), and

*TN*: the number of correct classifications of negative examples (**true negative**).

ACCURACY = (TP+TN) / (TP+FN+FP+TN)

# Precision and recall measures (cont…)

| | Classified Positive | Classified Negative |
|---|---|---|
| Actual Positive | TP | FN |
| Actual Negative | FP | TN |

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

- Precision *p (Predicted Positive Rate)* is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.
- Recall *r (True Positive Rate(TPR))* is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

# An example

| | Classified Positive | Classified Negative |
|---|---|---|
| Actual Positive | 1 | 99 |
| Actual Negative | 0 | 1000 |

- This confusion matrix gives
  - precision $p$ = 100% and
  - recall $r$ = 1%

  because we only classified one positive example correctly and no negative examples wrongly.
- Note: precision and recall only measure classification on the positive class.

# $F_1$-value (also called $F_1$-score)

- It is hard to compare two classifiers using two measures. $F_1$ score combines precision and recall into one measure

$$F_1 = \frac{2pr}{p + r}$$

$F_1$-score is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

- The harmonic mean of two numbers tends to be closer to the smaller of the two.
- For $F_1$-value to be large, both *p* and *r* much be large.

# Sensitivity and Specificity

- In statistics, there are two other evaluation measures:
  - Sensitivity: Same as Recall (TPR)
  - Specificity: Also called True Negative Rate (TNR)

- Then we have $$TNR = \frac{TN}{TN + FP}$$

$$FPR = 1 - specificity$$

| | Classified Positive | Classified Negative |
|---|---|---|
| Actual Positive | TP | FN |
| Actual Negative | FP | TN |

# Receive operating characteristics curve

- It is commonly called the ROC curve.
- It is a plot of the true positive rate (TPR) against the false positive rate (FPR).
- True positive rate:

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

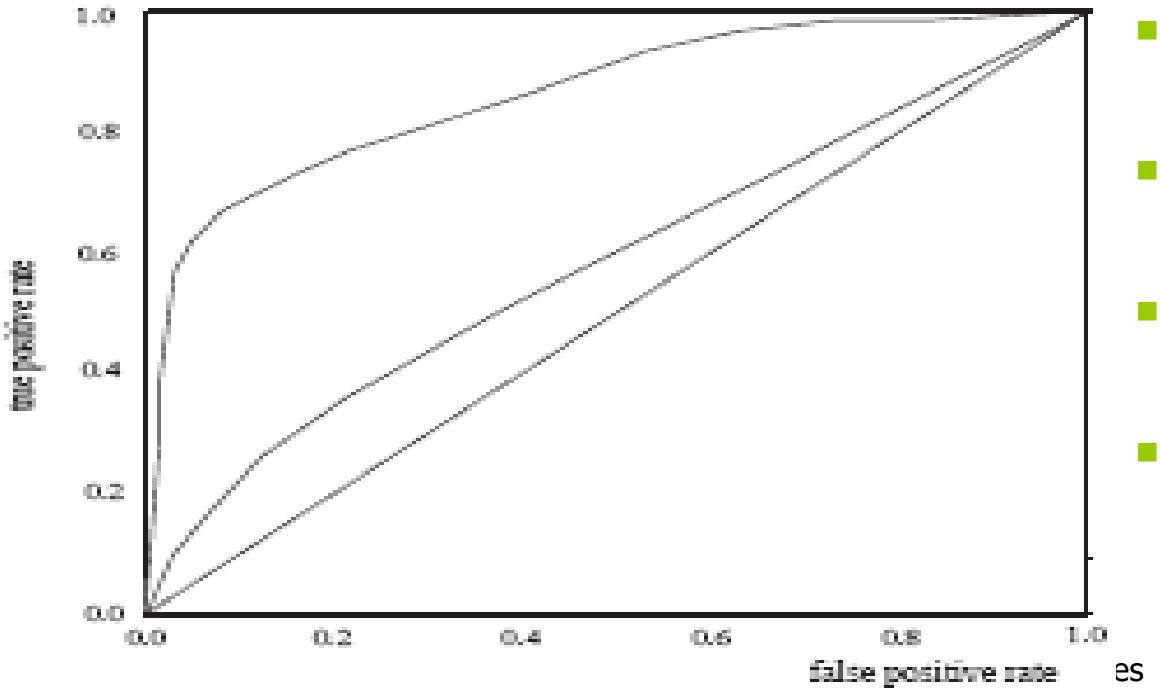
- False positive rate:

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

| | Classified Positive | Classified Negative |
|---|---|---|
| Actual Positive | TP | FN |
| Actual Negative | FP | TN |

# ROC Curves

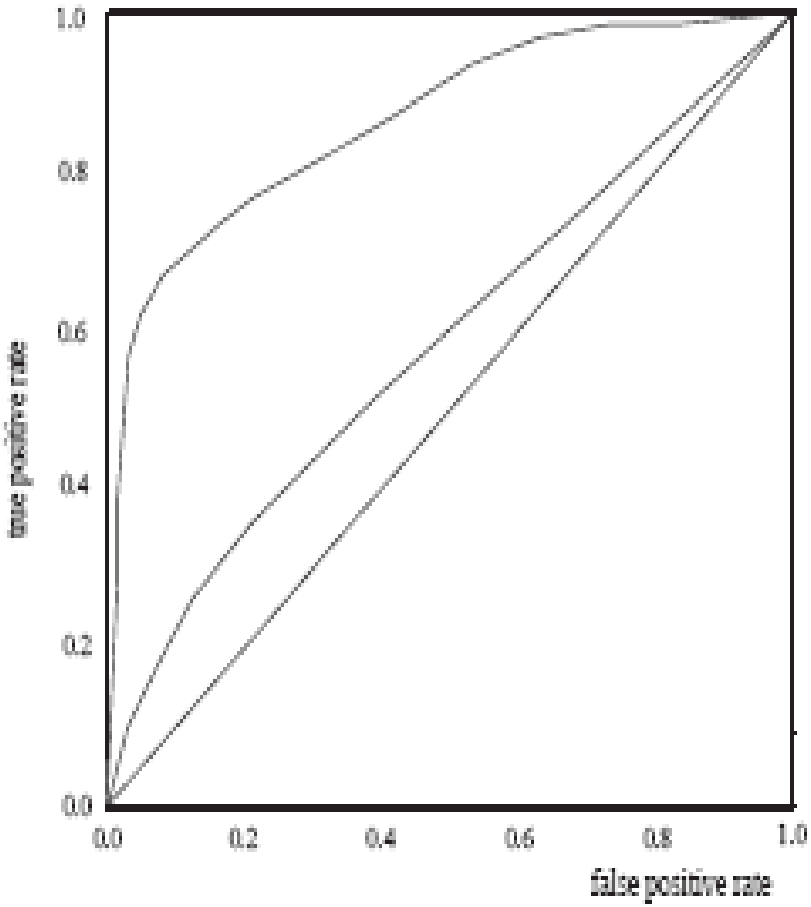
- ROC (Receiver Operating Characteristics) curves: for visual comparison of classification models
- Originated from signal detection theory
- Shows the trade-off between the true positive rate and the false positive rate




- Vertical axis represents the true positive rate
- Horizontal axis rep. the false positive rate
- The plot also shows a diagonal line
- A model with perfect accuracy will have an area of 1.0

# ROC Graphs

- The area under the ROC curve is a measure of the accuracy of the model
- Rank the test tuples in decreasing order: the one that is most likely to belong to the positive class appears at the top of the list
- The closer to the diagonal line (i.e., the closer the area is to 0.5), the less accurate is the model



# ROC Graphs

- **Features of ROC Graphs**
- An ROC curve or point is independent of class distribution or error costs (Provost et al., 1998).
- An ROC graph encapsulates all information contained in the confusion matrix, since *FN* is the complement of *TP* and *TN* is the complement of *FP* (Swets, 1988).
- ROC curves provide a visual tool for examining the tradeoff between the ability of a classifier to correctly identify positive cases and the number of negative cases that are incorrectly classified.

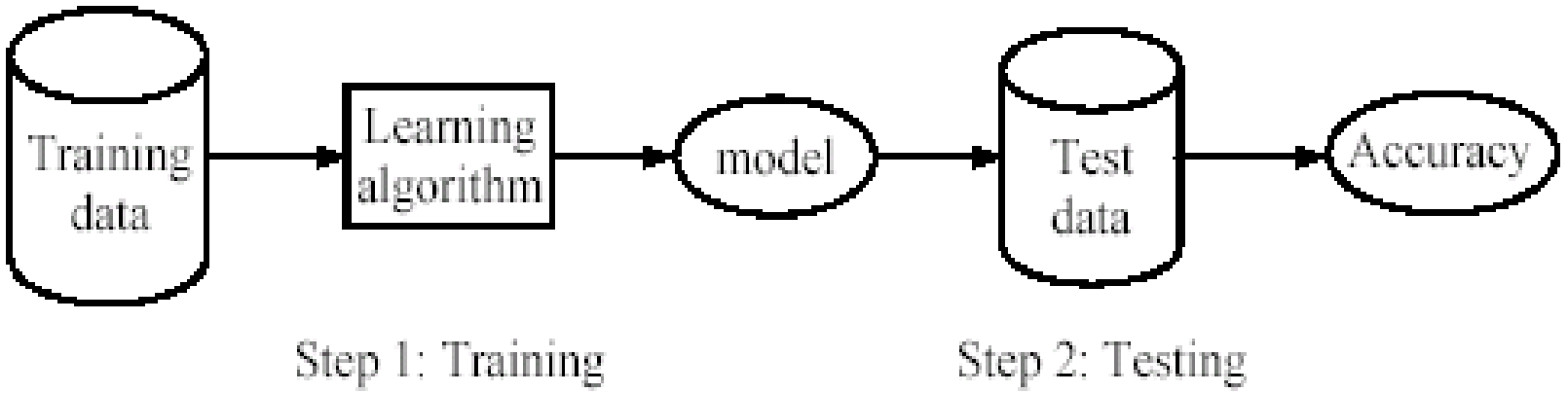
# Performance Evaluation: Methods for Spliting Data for Training and Testing

- Holdout
  - Reserve 2/3 for training and 1/3 for testing
  - Random sub-sampling (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
  - Sampling with replacement
- Training with **training**, **validation**, and **test sets**.

# Holdout Method

- Learning (training): Learn a model using the training data
- Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



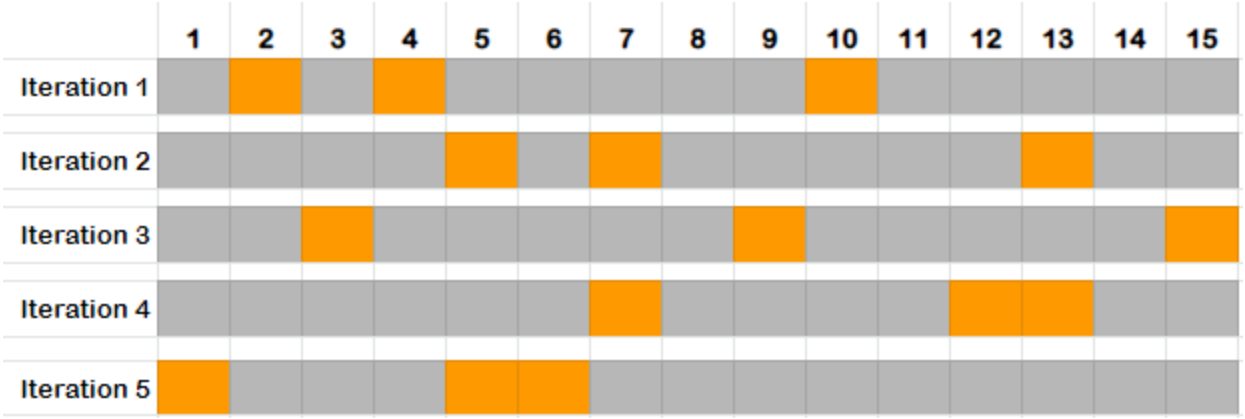
# Holdout method

- Holdout method
  - Given data is randomly partitioned into two independent sets (Usually: one third for testing, the rest for training )
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Problem: the samples might not be representative
    - Example: class might be missing in the test data
  - Advanced version uses stratification
    - Ensures that each class is represented with approximately equal proportions in both subsets

# Repeated holdout (Random sampling) method

- Random sampling: a variation of 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 calculated.
  - Repeat holdout k times, accuracy = avg. of the accuracies obtained

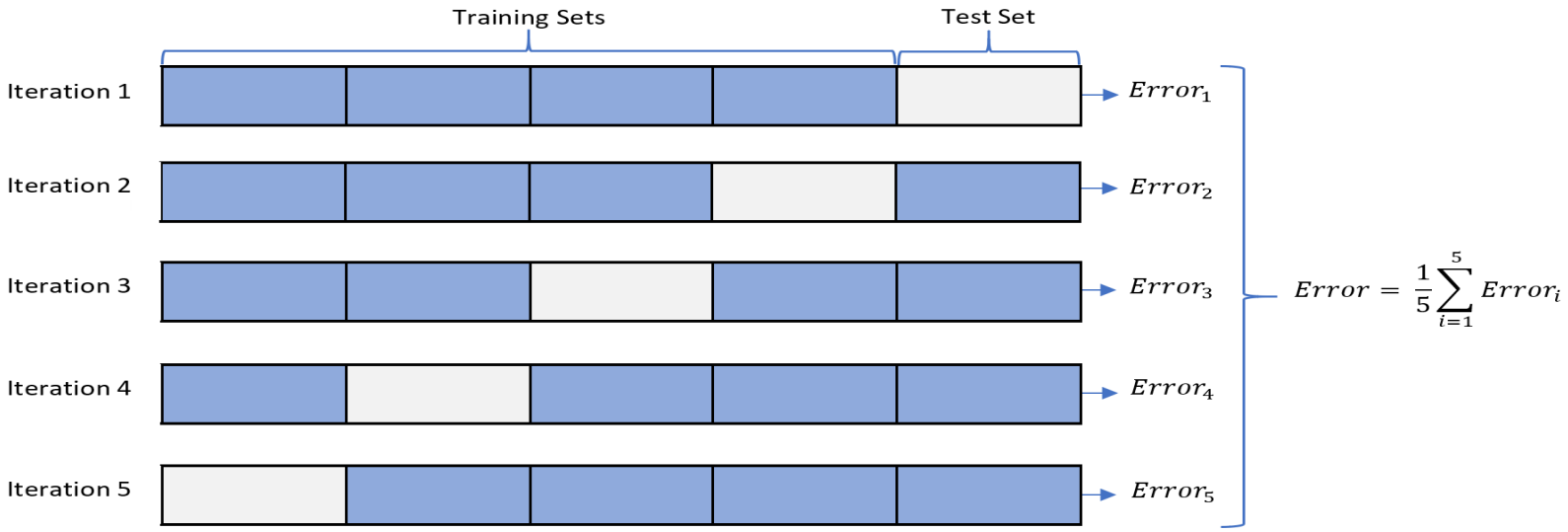
- Still not optimum: the different test sets overlap




$$\mathrm{acc_{cv}} = \sum_{i=1}^{k} \frac{\mathrm{acc_i}}{k}$$

# Cross-validation

- 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 the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate




K Fold CV, K=5

# More on cross-validation

- Standard method for evaluation: stratified ten-fold cross-validation
- Why ten? Extensive experiments have shown that this is the best choice to get an accurate estimate
  - There is also some theoretical evidence for this
- Stratification reduces the estimate's variance
- Even better: repeated stratified cross-validation
  - E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance)

# Leave-one-out cross validation

- Leave-one-out cross-validation is a particular form of cross-validation:
  - The number of folds is set to the number of training instances
  - i.e., a classifier has to be built n times, where n is the number of training instances
- Makes maximum use of the data
- No random subsampling involved
- Very computationally expensive (exception: NN)

# Bootstrap method

- Bootstrap
  - Works well with small data sets
  - Samples the given training tuples uniformly *with replacement*
    - i.e., each time a tuple is selected, it is equally likely to be selected again and re-added to the training set
- Several boostrap methods, and a common one is **.632 boostrap**
  - Suppose we are given a data set of d tuples. The data set is sampled d times, with replacement, resulting in a training set of d samples. The data tuples 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 (since $(1 - 1/d)^d \approx e^{-1} = 0.368$)
  - Repeat the sampling procedure k times, overall accuracy of the model:

$$acc(M) = \sum_{i=1}^{k} (0.632 \times acc(M_i)_{test\_set} + 0.368 \times acc(M_i)_{train\_set})$$

# Training with Validation Set

- **Validation set**: the available data is divided into three subsets,
  - a training set,
  - a validation set and
  - a test set.
- A validation set is used frequently for estimating parameters in learning algorithms.
- In such cases, the values that give the best accuracy on the validation set are used as the final parameter values.
- Cross-validation can be used for parameter estimating as well.

# Multiclass Classification with Binary Classifiers

- Classification involving more than two classes (i.e., > 2 Classes)
- Method 1. **One-vs.-all** (OVA): Learn a classifier one at a time
  - Given m classes, train m classifiers: one for each class
  - Classifier j: treat tuples in class j as *positive* & all others as *negative*
  - To classify a tuple **X**, the set of classifiers vote as an ensemble
- Method 2. **All-vs.-all** (AVA): Learn a classifier for each pair of classes
  - Given m classes, construct m(m-1)/2 binary classifiers
  - A classifier is trained using tuples of the two classes
  - To classify a tuple **X**, each classifier votes. X is assigned to the class with maximal vote
- Comparison
  - All-vs.-all tends to be superior to one-vs.-all
  - Problem: Binary classifier is sensitive to errors, and errors affect vote count

# Semi-Supervised Classification

- Semi-supervised: Uses labeled and unlabeled data to build a classifier
- Self-training:
  - Build a classifier using the labeled data
  - Use it to label the unlabeled data, and those with the most confident label prediction are added to the set of labeled data
  - Repeat the above process
  - Adv: easy to understand; disadv: may reinforce errors
- Co-training: Use two or more classifiers to teach each other
  - Each learner uses a mutually independent set of features of each tuple to train a good classifier, say $f_1$
  - Then $f_1$ and $f_2$ are used to predict the class label for unlabeled data X
  - Teach each other: The tuple having the most confident prediction from $f_1$ is added to the set of labeled data for $f_2$, & vice versa
- Other methods, e.g., joint probability distribution of features and labels

# Summary (I)

- Classification and regression are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends.
- Effective and scalable methods have been developed for classification: decision trees induction, Naive Bayesian classification, Bayesian belief network, rule-based classifier, Backpropagation, Support Vector Machine (SVM), nearest neighbor, neural networks, deep neural netwoks.
- Linear, nonlinear, and generalized linear models of regression can be used for regression Many nonlinear problems can be converted to linear problems by performing transformations on the predictor variables. Regression trees and model trees are also used for prediction.

# Summary (II)

- Stratified k-fold cross-validation is a recommended method for accuracy estimation. Significance tests and ROC curves are useful for model selection
- There have been numerous comparisons of the different classification and prediction methods, and the matter remains a research topic
- No single method has been found to be superior over all others for all data sets
- Issues such as accuracy, training time, robustness, interpretability, and scalability must be considered and can involve trade-offs, further complicating the quest for an overall superior method