Classification:
model development and evaluation

Classification

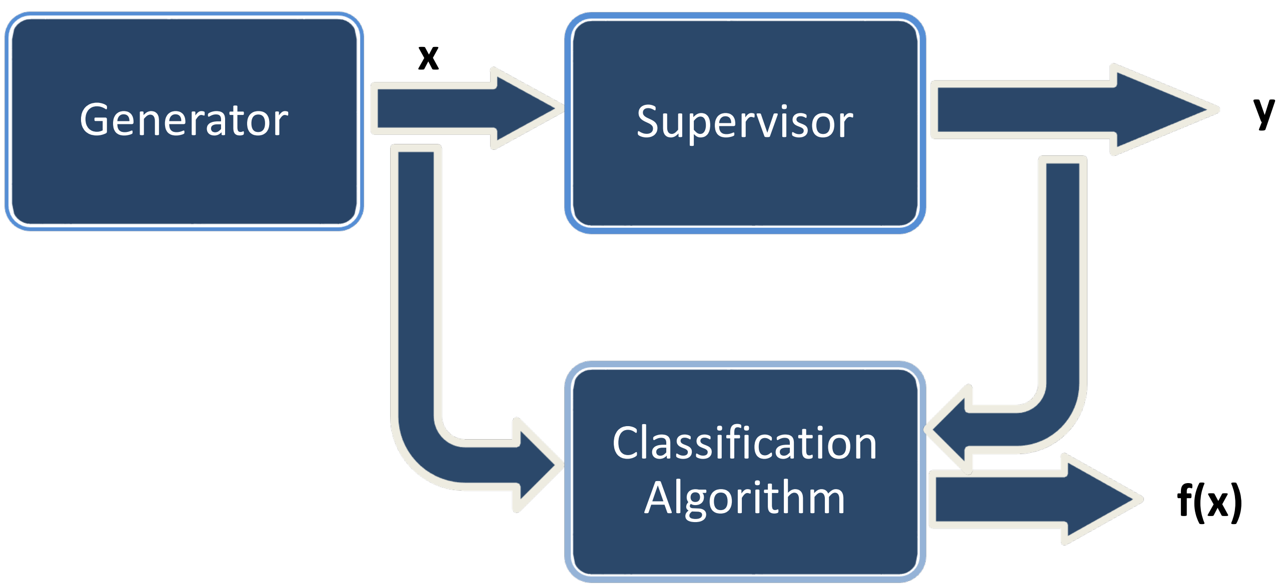
- Classification models: supervised learning methods for predicting value of a categorical target attribute.
- They generate a set of rules that allows the target class of future examples to be predicted.
- Theoretical viewpoint: classification algorithm development represents a fundamental step in emulating inductive capabilities of the human brain.
- Practical viewpoint: applicable in many different domains such as selection of target customers for a marketing campaign, fraud detection, image recognition, early diagnosis of disease, text cataloguing and spam email recognition.

Classification problems

- We have a data set $\mathbb{D}$ containing m observations described in terms of n explanatory attributes (predictive variables) and a categorical target attribute (a class or a label).
- The observations are also termed *examples, instances, data samples, records, data points*.
- Binary classification: the instances belong to two classes only.
- Multi-class or multi-category classification: there are more than two classes in the data set.
- A classification problem consists of defining an appropriate space $\mathbb{F}$ and an algorithm $A_{\mathbb{F}}$ that idenfifies a function $f^* \in \mathbb{F}$ that optimally describes the relationship between the predictive attributes and the target class.
- $\mathbb{F}$ is a class of functions $f(\mathbf{x})$: $\mathbb{R}^n \Rightarrow \mathbb{H}$ called hypotheses that represent hypothetical relationship of dependence between y_i and $\mathbf{x}_i$.
- $\mathbb{H}$ could be {0,1} or {-1,1} for a binary classification problem.

Components of a classification problem

- Generator: extract data example/instance **x**.
- Supervisor: for each **x**, return the value of the target class.
- Classification algorithm (or simply classifier) choses a function f from the hypothesis space to minimize a loss function.



Development of a classification model

Three main phases:

1. **Training phase.**

- the classification algorithm is applied to the examples belonging to a subset $\mathbb{T}$ of the data set $\mathbb{D}$.
- $\mathbb{T}$ is called the training data set.
- Classification rules are derived to allow users to predict a class to each observation **x.**

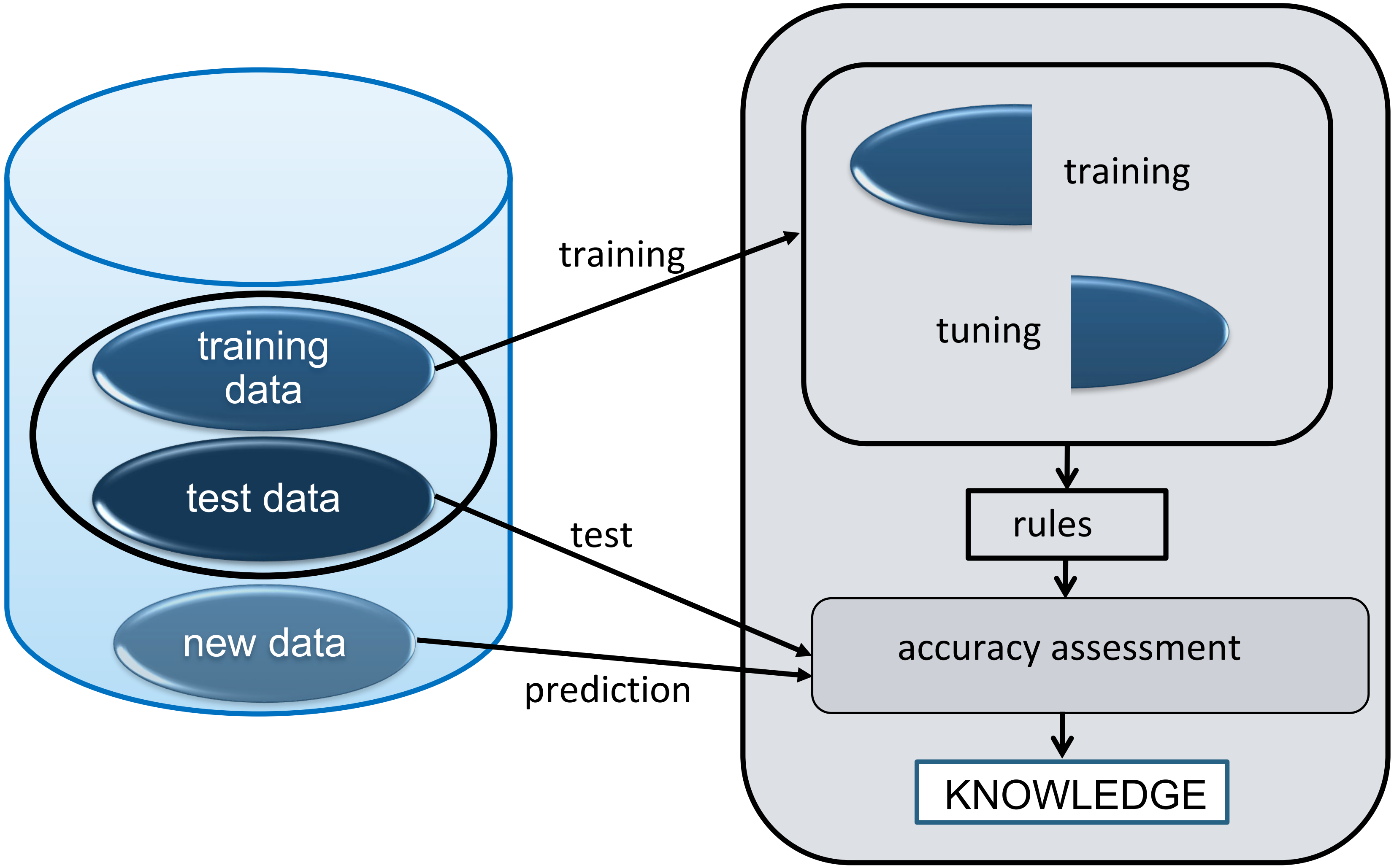
2. **Test phase.**

- The rules generated in the training phase are used to classify observations in $\mathbb{D}$ but not in $\mathbb{T}$.
- Accuracy is checked by comparing the actual target class with the predicted class for all instances in $\mathbb{V} = \mathbb{D} - \mathbb{T}$.
- Obvervations in $\mathbb{V} = \mathbb{D} - \mathbb{T}$ form the test set. The training and test sets are disjoint: $\mathbb{V} \cap \mathbb{T} = \emptyset$

3. **Prediction phase.**

- The actual use of the classification model to assign target class to completely new observations.
- This is done by applying the rules generated during the training phase to the attributes of the new instances.

Development of a classification model



Taxonomy of classification models

1. **Heuristic models**

   - Classification is achieved by applying simple and intuitive algorithms.
   - Example: *k-nearest neighbor method* based on distance between observations.
   - Another example: *classification trees* which apply divide-and-conquer technique to obtain groups of samples that are as homogenous as possibles with respect to the target variables.

2. **Separation models**

   Divide the attribute space into H distinct regions.

   - All observations in a region are assigned the same class.
   - How to determine these regions? Not too complex or many, not too simple or few either.
   - Define a loss function to take into account the misclassificied points and applied an optimization algorithm to derive a subdivision into regions that minimizes the total loss.
   - Examples: *discriminant analysis, perceptron methods, neural networks, support vector machines, classification trees.*

Taxonomy of classification models

3. Regression models

- *Logistic regression* is an extension of linear regression suited to handling binary classification problems.
- Main idea: convert binary classification problem via a proper transformation into a linear regression problem.

4. Probabilistic models

- A hypothesis is formulated regarding the functional form of the conditional probabilities $P_{\mathbf{x}|y}(\mathbf{x}|y)$ of the observations given the target class. This is known as class-conditional probabilities.
- Based on an estimate of the prior probabilities $P_y(y)$ and using Bayes‘ theorem, calculate the posterior probabilities $P_{y|\mathbf{x}}(y|\mathbf{x})$ of the target class.
- Example: *Naive Bayes classifiers and Bayesian networks*.

Evaluation of a classification model

1. **Accuracy**

- A proportion of the observations that are correctly classified by the model.
- Usually one is more interested in the accuracy of the model on the test data set $\mathbb{V}$
- Let $L(y_i, f(\mathbf{x_i})) = 1$ if $y_i \neq f(\mathbf{x_i})$; 0 otherwise.

  Then

  $$acc_A(\mathbb{V}) = 1 - (1/v) \sum_{i=1}^{v} L(y_i, f(\mathbf{x_i}))$$

  Similarly

  $$error_A(\mathbb{V}) = 1 - acc_A(\mathbb{V}) = (1/v) \sum_{i=1}^{v} L(y_i, f(\mathbf{x_i}))$$

  where v is the number of samples in the test data set $\mathbb{V}$,

  A is the learning algorithm.

- Note: it could also be of interest to report the accuracy and the error on the training data set $\mathbb{T}$.

Evaluation of a classification model

2. **Speed**
   - Long computation time on large data sets can be reduced by means of random sampling scheme.
3. **Robustness**
   - The method is robust if the classification rules generated and the corresponding accuracy do not vary significantly as the choice of training data and test data sets varies.
   - It must also be able to handle missing data and outliers well.
4. **Scalibility**
   - Able to learn from large data sets.
5. **Interpretability**
   - Generated rules should be simple and easily understood by knowledge workers and domain experts.

Evaluation of a classification model

Holdout method

- Divide the available m observations in the data set $\mathbb{D}$ into training data set $\mathbb{T}$ and test data set $\mathbb{V}$.

- The t observations in $\mathbb{T}$ is usually obtained by random selection.

- The number of observations in $\mathbb{T}$ is suggested to be between one half and two thirds of the total number of observations in $\mathbb{D}$.

- The accuracy of the classification algorithm via the holdout method depends on the test set $\mathbb{V}$.

- In order to better estimate this accuracy, different strategies have been recommended.

Evaluation of a classification model

Repeated random sampling

- Simply replicate the holdout method r times.

- For each repetition k = 1,2, ..., r :

  - A random training data set $\mathbb{T}_k$ having t observations is generated.

  - Compute $acc_{AF}(\mathbb{V}_k)$, the accuracy of the classifier on the corresponding test set $\mathbb{V}_k$, where

    $\mathbb{V}_k = \mathbb{D} - \mathbb{T}_k$.

- Compute the average accuracy:

$$acc_A = (1/r) \sum_{k=1}^{r} acc_{AF}(\mathbb{V}_k)$$

- Drawback: no control over the number of times each observation may appear, outliers may cause undesired effects on the rules generated and the accuracy.

Evaluation of a classification model

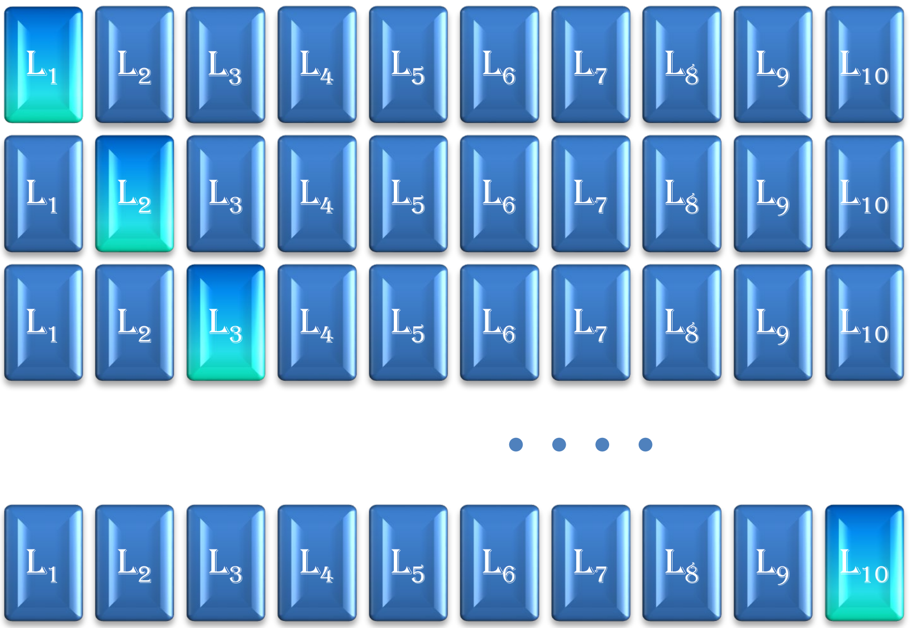
Cross-validation

- Divide the data into r disjoint subsets, $\mathbb{L}_1, \mathbb{L}_2, \ldots \mathbb{L}_r$ of (almost) equal size.
- For iterations $k = 1,2, \ldots r$
  - Let the test set be $\mathbb{V}_k = \mathbb{L}_k$
  - And the training $\mathbb{T}_k = \mathbb{D} - \mathbb{L}_k$.
  - Compute $acc_{AF}(\mathbb{V}_k)$
- Compute the average accuracy:

$$acc_A = (1/r) \sum_{k=1}^{r} acc_{AF}(\mathbb{V}_k)$$

- Usual value for r is $r = 10$

(ten-fold cross-validation)



Evaluation of a classification model

Leave-one-out

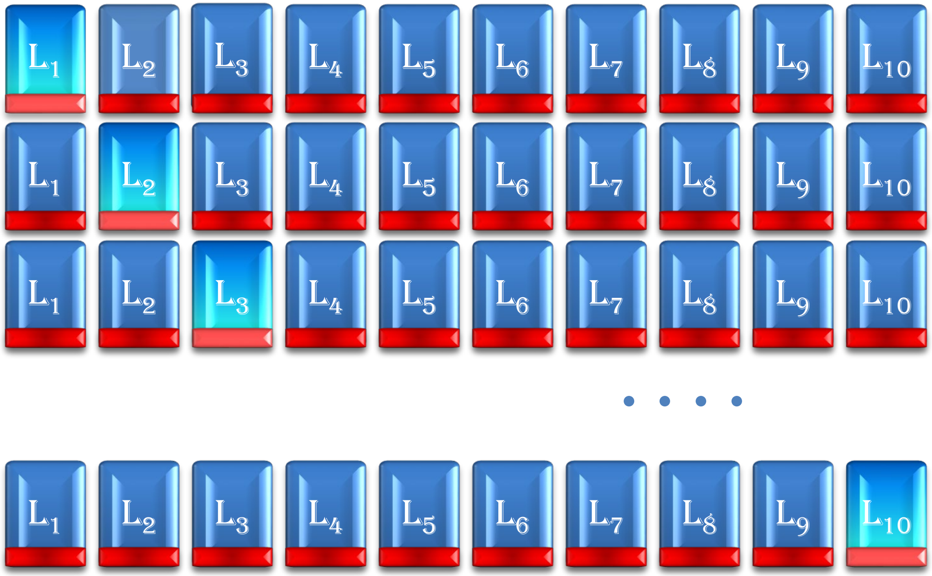
- Cross-validation method with the number of iterations *r* is set to *m*.

- This means each of the *m* test sets consists only of 1 sample and the corresponding training data set consists of *m-1* samples.

Note: Instead of random sampling to partition the data set $\mathbb{D}$ into training set $\mathbb{T}$ and test set V, stratified random sampling could be used to ensure that the proportion of observations belonging to each target class is the same in both $\mathbb{T}$ and $\mathbb{V}$.

Blue: Class 0

Red: Class 1



Evaluation of a classification model

Confusion matrix

- In many situations, just computing the accuracy of the classifier may not be enough.
- Example 1: Medical domain.
  - The value of 1 means the patient has a given medical condition, -1 means he does not.
  - If only 2% of all patients in the data base have the condition, then we achieve accuracy rate of 98% by having the rule „the patient does not have the condition“.
- Example 2: Customer retention.
  - The value of 1 means the customer has cancelled the service, 0 means the customer is still active.
  - If only 2% of the available data correspond to customers who have cancelled the service, the simple rule „the customer is still active“ has an accuracy rate of 98%.

Evaluation of a classification model

Confusion matrix for a binary target attribute encoded with the class values {-1,+1}

		predictions		
		-1 (negative)	+1 (positive)	total
examples	-1 (negative)	p	q	p+q
	+1 (positive)	u	v	u+v
	total	p+u	q+v	m

- **Accuracy:** among all samples, what is the proportion that is correctly predicted?

  $acc = (p+v)/(p+q+u+v) = (p+v)/m$

- **True negative rate:** among all negative examples, proportion of correct predictions is **specificity =**

  $tnr = p/(p+q)$

- **False positive rate:** among all negative examples, proportion of incorrect predictions is **the false alarm rate =**

  $fpr = q/(p+q) = 1 - tn$

- **True positive rate:** among all positive examples, proportion of correct predictions is **recall = sensitivity** $= tpr = v/(u+v)$

- **False negative rate:** among all positive examples, proportion of incorrect prediction is $fnr = u/(u+v) = 1 - tpr$

Evaluation of a classification model

<u>Confusion matrix for a binary target attribute encoded with the class values {-1,+1}</u>

		predictions		
		-1 (negative)	+1 (positive)	total
examples	-1 (negative)	p	q	p+q
	+1 (positive)	u	v	u+v
	total	p+u	q+v	m

- **Precision:** among all positive predictions, the proportion of actual positive samples is

  prc = v/(q + v)

- **Geometric mean** = **gm1 = sqrt(tpr × prc)**
- **Geometric mean** = **gm2 = sqrt(tpr × tnr)**

- **F-measure** = **{(β² + 1)tpr × prc}/(β² × prc + tpr)** where β > 0.
- If β = 1, F-measure = 2 (tpr × prc)/(prc + tpr)

$$= \frac{1}{\frac{1}{2}\left(\frac{1}{tpr}+\frac{1}{prc}\right)}$$ = harmonic mean of precision and tpr(recall).

Evaluation of a classification model

Confusion matrix for a binary target attribute encoded with the class values {-1,+1}

Example:

- **66 financial institutions are classified as either solvent (Event/Class = +1) or bankrupt (Non-Event/Class = -1)**

based on 2 financial ratios: x_1 and x_2

- **37 Solvent, 29 Bankrupt.**
- **Logistic regression model:** $P(Y=1|x_1,x_2) = 1/(1 + \exp(5.9798 - 0.285\ x_1 - 4.5361\ x_2))$
- **Output from SAS (partial):**

Prob Level	Correct Event	Correct Non-Event	Incorrect Event	Incorrect Non-Event	Correct	Sensi-tivity	Speci-ficity
0.000	37	0	29	0	56.1	100.0	0.0
0.020	37	8	21	0	68.2	100.0	27.6
0.040	36	11	18	1	71.2	97.3	37.9
0.060	36	13	16	1	74.2	97.3	44.8
0.080	36	15	14	1	77.3	97.3	51.7
0.100	36	17	12	1	80.3	97.3	58.6
0.120	36	20	9	1	84.8	97.3	69.0
0.140	35	21	8	2	84.8	94.6	72.4
0.160	35	21	8	2	84.8	94.6	72.4
0.180	35	22	7	2	86.4	94.6	75.9

Evaluation of a classification model

Confusion matrix for a binary target attribute encoded with the class values {-1,+1}

Example:

Prob Level	Correct Event	Correct Non-Event	Incorrect Event	Incorrect Non-Event	Correct	**Recall/TPR** Sensitivity	**TNR** Specificity
0.000	37	0	29	0	56.1	100.0	0.0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0.120	36	20	9	1	84.8	97.3	69.0
0.140	35	21	8	2	84.8	94.6	72.4
0.160	35	21	8	2	84.8	94.6	72.4
0.180	35	22	7	2	86.4	94.6	75.9

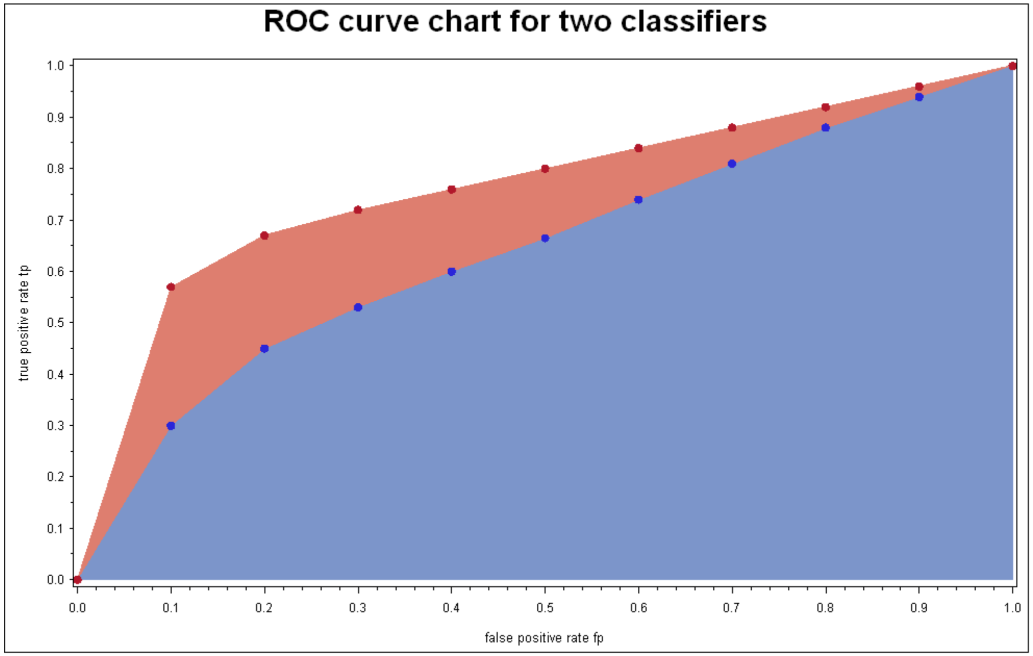
- Given x_1 and x_2, compute $P(Y=1|x_1,x_2)$.
- If $P \geq$ ProbLevel, predict event/solvent. Otherwise, predict non-event/bankrupt
- When ProbLevel = 0, all samples are predicted as event (Solvent).
- All 37 Solvent banks are correctly predicted, all 29 Bankrupt banks are incorrectly predicted.

When ProbLevel = 0.18:

- 35 Solvent banks are correctly predicted, 22 Bankrupt banks are correctly predicted.
- % Sensitivity = tpr = 35/37 = 94.6%, % Specificity = tnr = 22/29 =75.9%.
- % Correct = (35+22)/66 = 86.4%

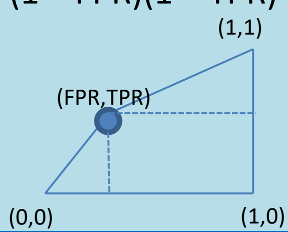
Evaluation of a classification model

ROC (receiver operating characteristics) curve charts



Area under the curve:

½ (TPR)(FPR) + (1 – FPR)(TPR) + ½ (1 – FPR)(1 – TPR)
= ½ (1 – FPR) + ½ TPR
= ½ (TNR + TPR)

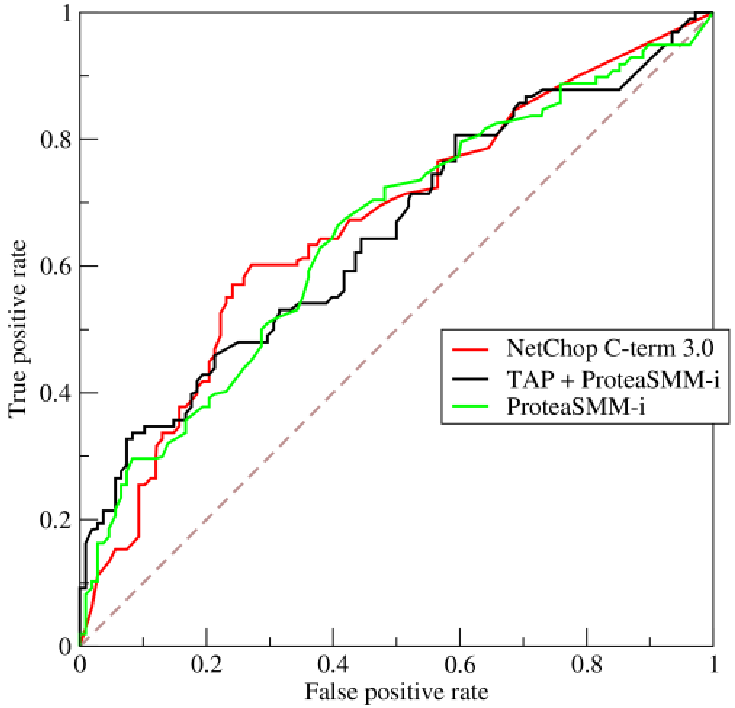


- Two dimensional plot, fp on the horizontal axis, tp on the vertical axis.
- The point (0,1) represents the **ideal classifier**.
- The point (0,0) corresponds to a classifier that predicts class {-1} for all samples.
- The point (1,1) corresponds to a classifier that predicts class {1} for all samples.
- Parameters in a classifier may be adjusted so that tp can be increased, but at the same time increasing fp.
- A classifier with no parameters to be (further) tuned yields only 1 point on the chart (FPR,TPR).
- The **area** beneath the ROC provides means to compare the accuracy of various classifiers.
- The ROC curve with the greatest area is preferable.

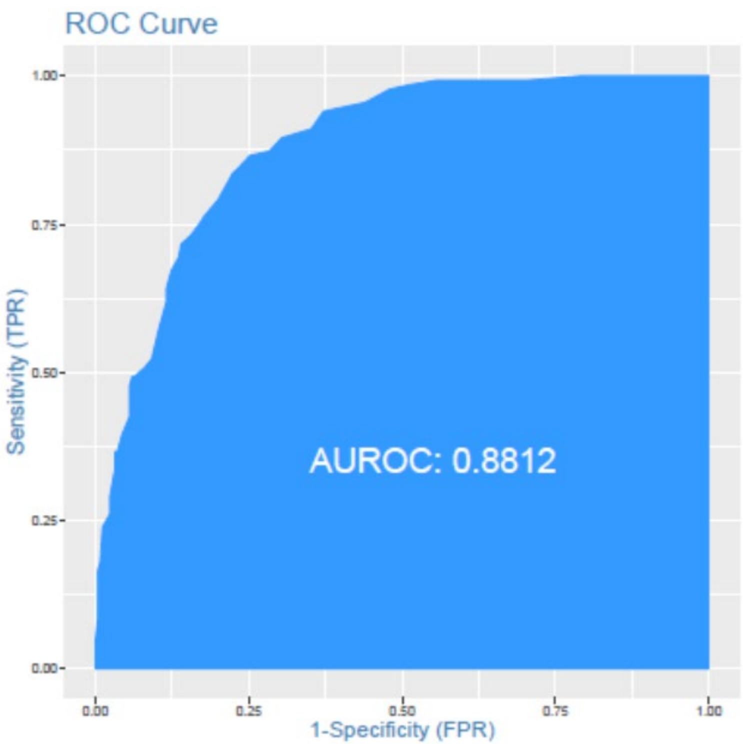
Evaluation of a classification model

ROC (receiver operating characteristics) curve charts: more examples.

From Wikipedia.



Output from R.



Evaluation of a classification model

Cumulative gain chart

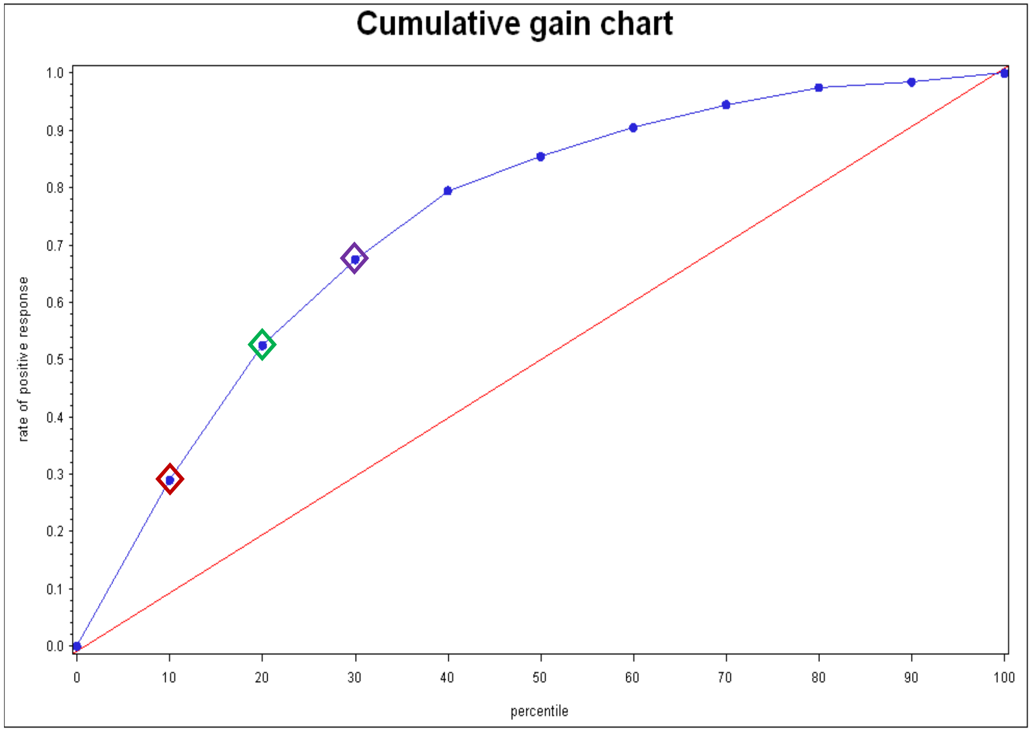
Number of customers contacted	Number of positive response
0	0
100000	5800
200000	10500
300000	13500
400000	15900
500000	17100
600000	18100
700000	18900
800000	19500
900000	19700
1000000	20000

- A company has a total of m = 1000000 customers.
- Based on past campaigns, the proportion of customers who might respond to the promotion is 2%.
- If we select a random sample of s customers, 0.02s customers are expected to respond.
- Can we do better than this?
- A classifier with a score function can help:
  - Score the customers and rank these scores, from the highest to the lowest.
  - For each *s*, consider the set $\mathbb{S}$ consisting only the first *s* customers on the ranked list.

Evaluation of a classification model

- Cumulative gain for the classifier is shown below:

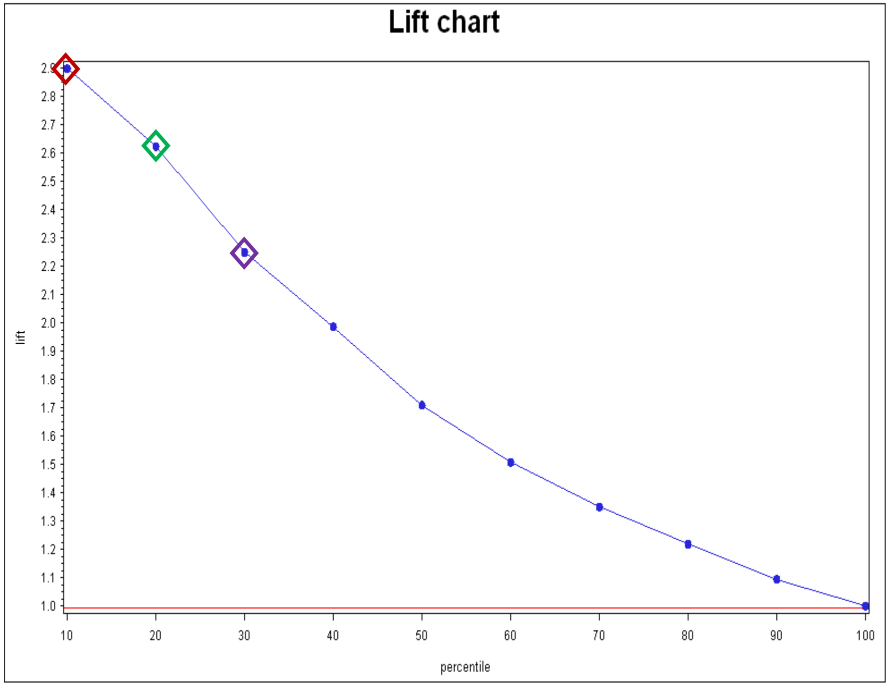
Number of customers contacted	Number of positive response
0	0
100000	5800
200000	10500
300000	13500
400000	15900
500000	17100
600000	18100
700000	18900
800000	19500
900000	19700
1000000	20000



◇ 5800/20000 = 0.29
◇ 10500/20000 = 0.525
◇ 13500/20000 = 0.675

Evaluation of a classification model

Lift chart



$a/m = 20000/1000000 = 0.02 = 2\%$

◇ $(5800/100000)/0.02 = 2.9$

◇ $(10500/200000)/0.02 = 2.625$

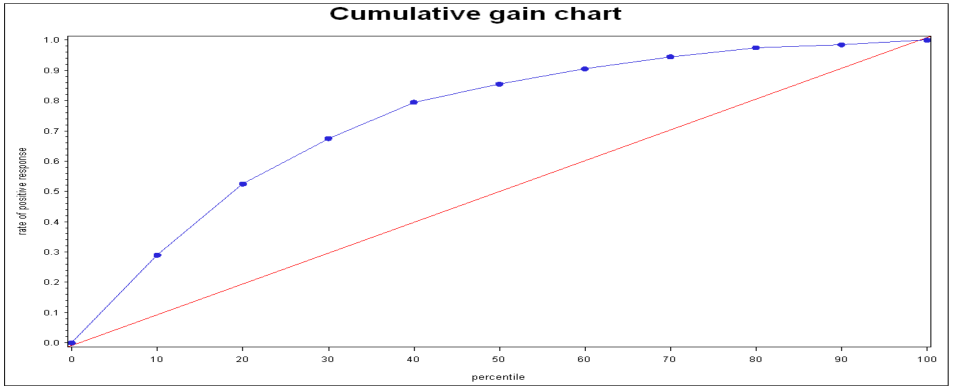
◇ $(13500/300000)/0.02 = 2.25$

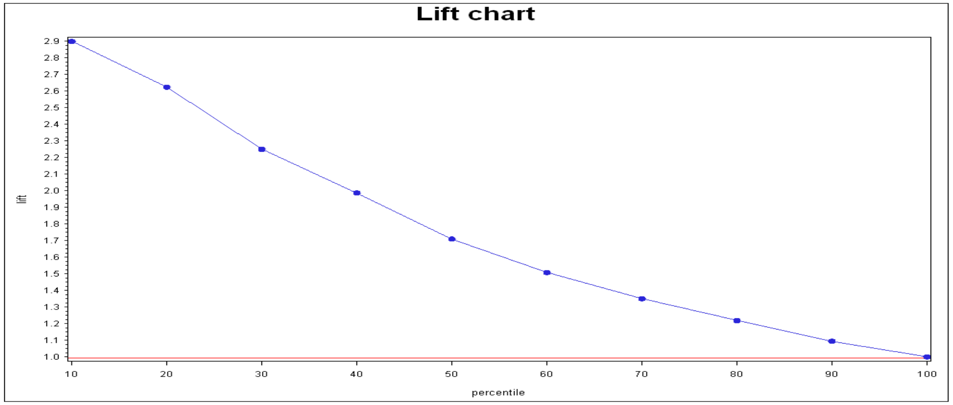
- The lift measures the accuracy based on the density of positive observations inside the set that has been identified based on model predictions.
- Let
  - $\mathbb{S}$ be a subset of observations of interest.
  - s: the number of observations in $\mathbb{S}$.
  - b/s: proportion of positive observations in S.
- Let $\mathbb{D}$ be the entire dataset having m observations and a/m be the proportion of positive observations in $\mathbb{D}$.

lift = (b/s)/(a/m)

Evaluation of a classification model

Use of Cumulative Gain and Lift charts





- **Cumulative gain chart:** a greater area corresponds to classification method that is more effective overall.
- **Lift chart:** maximum lift at a specific value *s* on the horizontal axis that indicates the actual number of recipients of the marketing campaign determined according to the available budget. Classification method is selected based on the maximum lift at a specific value *s* on the horizontal axis.

Evaluation of a regression model

Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE):

$$\mathrm{RMSE} = \sqrt{\frac{\sum_p (\tilde{y}_p - y_p)^2}{N}}$$

$$\mathrm{MAE} = \frac{\sum_p |\tilde{y}_p - y_p|}{N}$$

N : the number of samples

$\tilde{y}_p$: the predicted value for sample $p = 1, 2, \ldots N$

y_p : the actual (target) value of sample $p = 1, 2, \ldots N$

$\overline{y}$: the average value of y_p

Relative Root Mean Squared Error (RRMSE) and the Relative Mean Absolute Error (RMAE):

$$\mathrm{RRMSE} = 100 \times \sqrt{\sum_p (\tilde{y}_p - y_p)^2 / \sum_p (\overline{y} - y_p)^2}$$

$$\mathrm{RMAE} = 100 \times \sum_p |\tilde{y}_p - y_p| / \sum_p |\overline{y} - y_p|$$

When there are more than 2 classes

- The article “Multiclass cancer classification using a feature subset-based ensemble from microRNA expression profiles” by Y. Piao, M. Piao and K.H. Ryu, Computers in Biology and Medicine 80 (2017) – 39—44 presents a performance comparison of a feature subset-based ensemble method versus C4.5 and Support Vector Machines.
- Dataset: 4 classes, 87+27+34+67 = 215 samples, # features = 1047.

Dataset	Diseases	Samples	miRNAs
D_1	BRCA	87	1,047
	DLBC	27	
	PAAD	34	
	PRAD	67	

When there are more than 2 classes

Ensemble learning:

- Each classifier (C4.5 DT or SVM) in the ensemble in trained using a different feature subset.
- The average posteriori probability is used to combine the prediction of each classifier in the ensemble.
- Experimental setting: 10 fold cross-validation and leave-one-out (50 runs each).
- Number of classifiers in the ensemble: 20
- Evaluation metric:
  - Accuracy
  - Sensitivity = # true positives/(# true positives + # false negative)
  - Specificity = # true negatives/(#true negatives + # false positive)
  - AUC

When there are more than 2 classes

Feature selection:

- $IG(X|Y) = H(X) - H(X|Y)$
- $SU(X,Y) = 2 \times IG(X|Y)/(H(X) + H(Y))$

- H(X) and H(Y) are the entropy values of variables X and Y
- IG(X|Y) is the information gain of X after observing variable Y
- SU(X,Y) has values in [0,1] where 1 indicates complete correlation and 0 indicates no correlation. SU = Symmetrical Uncertainty.
- Relevant feature: A feature X is relevant if the SU value to the class SU(X,C) is larger than a user-predefined threshold. Note: here we let Y = C.
- Redundant feature: Relevant features X and Y are redundant if SU(X,Y) is larger than min(SU(X,C),SU(Y,C))
- R implementation.

When there are more than 2 classes

Results :

Classification results on D_1 (C4.5 as the base classifier).

	10-fold cross validation			Leave-one-out cross validation		
	recall/tpr Sensitivity	tnr Specificity	AUC	Sensitivity	Specificity	AUC
BRCA	0.977	0.977	0.992	0.977	0.961	0.995
DBLC	0.936	1	0.981	0.963	1	0.981
PAAD	0.941	0.983	0.994	0.912	0.978	0.98
PARD	0.955	0.986	0.988	0.925	0.986	0.988
Overall	0.963	0.984	0.99	0.949	0.976	0.989

Classification results on D_1 (SVM as the base classifier).

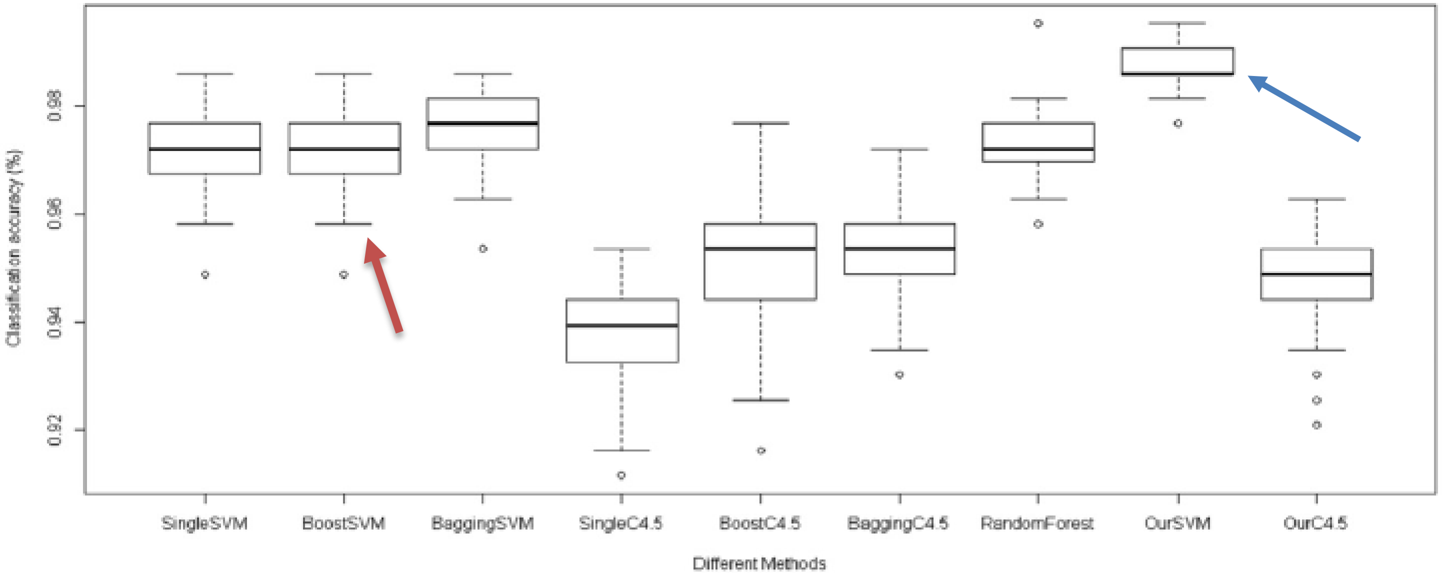
	10-fold cross validation			Leave-one-out cross validation		
	Sensitivity	Specificity	AUC	Sensitivity	Specificity	AUC
BRCA	0.977	1	0.993	0.977	0.992	0.988
DBLC	1	1	1	1	1	1
PAAD	1	0.989	0.994	0.971	0.994	0.996
PARD	1	1	1	1	0.993	1
Overall	0.991	0.998	0.996	0.962	0.994	0.994

Overall = simple/weighted average (by number of samples)

When there are more than 2 classes

Results :

- Boxplot:



- Test the hypothesis:

- H_0: mean accuracy of the new method = mean accuracy of an old method
- H_a: mean accuracy of the new method ≠ mean accuracy of an old method

H_0 is rejected with t = - 10.186 and p-value = 0.000

Reference

Business Intelligence: Data Mining and Optimization for Decision Making by Carlo Vercellis, 2009, Wiley. Chapters 10.1 and 10.2.

Also available in RBR Section Central Library.