

# Improving stacking methodology for combining classifiers: applications to cosmetic industry

Charles Gomes, Hisham Nocairi, Marie Thomas, Gilbert Saporta

▶ **To cite this version:**

Charles Gomes, Hisham Nocairi, Marie Thomas, Gilbert Saporta. Improving stacking methodology for combining classifiers: applications to cosmetic industry. Electronic Journal of Applied Statistical Analysis, 2016, 09 (2), pp.340 - 361. 10.1285/i20705948v9n2p340 . hal-02471754

**HAL Id: hal-02471754**
**https://cnam.hal.science/hal-02471754**

Submitted on 9 Feb 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License




**Electronic Journal of Applied Statistical Analysis**
**EJASA, Electron. J. App. Stat. Anal.**
http://siba-ese.unisalento.it/index.php/ejasa/index
e-ISSN: 2070-5948
DOI: 10.1285/i20705948v9n2p340

**Improving stacking methodology for combining classifiers: applications to cosmetic industry**
By Noçairi et al.

Published: 14 October 2016

This work is copyrighted by Università del Salento, and is licensed under a Creative Commons Attribuzione - Non commerciale - Non opere derivate 3.0 Italia License.
For more information see:
http://creativecommons.org/licenses/by-nc-nd/3.0/it/

*Electronic Journal of Applied Statistical Analysis*
Vol. 09, Issue 02, October 2016, 340-361
DOI: 10.1285/i20705948v9n2p340

# Improving stacking methodology for combining classifiers: applications to cosmetic industry

Noçairi H.*[a], Gomes C.[a], Thomas M.[a], and Saporta G.[b]

[a] *L'Oréal Recherche,1 avenue Eugène Schueller, BP22, 93601 Aulnay sous bois, France*
[b] *CEDRIC, CNAM , 292 rue Saint Martin, 75141 Paris cedex 03, France*

Published: 14 October 2016

Stacking (Wolpert, 1992; Breiman, 1996) is known to be a successful way of linearly combining several models. We modify the usual stacking methodology when the response is binary and predictions highly correlated, by combining predictions with PLS-Logistic Regression instead of ordinary least squares. For small data sets we develop a strategy based on repeated split samples in order to select relevant variables and ensure the reproductibility of the final model. Five base (or level-0) classifiers are combined in order to get an improved rule which is applied to a classical benchmark of UCI Machine Learning Repository. Our methodology is then applied to the prediction of dangerousness of 165 chemicals used in the cosmetic industry, described by 35 *in vitro* and *in silico* characteristics, since faced to safety constraints, one cannot rely on a single prediction method, especially when the sample size is low.

**keywords:** Stacking, PLS-DA, Boosting, Naive Bayes, SVM, Safety evaluation.

## 1 Introduction

The data analytical strategy presented in this paper has been motivated by the following industrial context: L'ORÉAL wished to have an alternative approach to the animal experimentation which is now banned, in order to estimate the safety of its chemicals

*Corresponding author: hnocairi@rd.loreal.com.

©Università del Salento
ISSN: 2070-5948
http://siba-ese.unisalento.it/index.php/ejasa/index

coded as a binary response with two categories: "danger" or "no danger" using using all chemical's information.

Many supervised classification techniques are available in the statistical literature, belonging to various families of prediction models which handle the problem from different ways (linear, non-linear, probabilistic). In this paper, which extends Gomes et al. (2012), we decided to select 5 methods representing 5 broad families of statistical models; though very different, all these models provide membership probabilities.

In order to avoid any bias induced by the use of a specific statistical method, we have preferred to use an appropriate combination of all of them. In other words, the methodology used can be seen as a battery of statistical methods and instead of selecting the best one in terms of AUC (Area Under Curve), misclassification rate or whatsoever, the solution consists in combining their predictions with optimally chosen weights, which leads to a meta-model.

This meta-model is obtained by the stacking methodology of Wolpert (1992) and Breiman (1996). Stacking has been chosen since it needs no specific assumption unlike Bayesian Model Averaging, is very simple for it is a linear combination of base or level 0 models. Stacking proved its efficiency in many contexts and machine learning competitions such as the Netflix prize (http://www.netflixprize.com/) under the name of "blending". For a recent review cf (Sesmero et al., 2015).

We bring the following improvement to stacking: weights are found thanks to a PLS (Partial Least Square) logistic regression since the outcome is categorical and the predictions highly correlated.

The paper is organized as follows: After a short survey of the main approaches to combine predictors (section 2), we focus on stacking for binary classification (section 3). In section 4 we present the five classifiers we use in applications, with an experiment on a data set from the UCI Machine Learning repository. In section 5, we develop an heuristic process for small data sets aiming at variable selection, based on cross-validation, avoiding that some categories become empty. Section 6 presents the application to the industrial context of cosmetics. We conclude by some perspectives.

## 2 Combining predictions: a short overview

Let us consider the following situation of predicting a response variable $y$ (continuous or binary) with the help of $p$ predictors, either continuous or categorical. When one does not know the generative true model of the data, it is a common use, especially in machine learning or pattern recognition to try several predictive algorithms $\hat{y}_m = f_m(x)$ $m = 1, ..., M$ each algorithm is called a base or level-0 *learner*.

Given some criterion, (eg the sum of squared residuals or $R^2$ for a continuous response, the AUC for a binary response), we may choose the best model, provided we compute the criterion on a validation set or with some cross validation technique, since the more complex model will fit the best on the training data and there is no guarantee of generalization.

Instead of choosing the best algorithm among the $M$ in competition, we may also

combine them in order to get a better prediction model.

There are several ways of combining models (see Hastie et al., 2009, Kuncheva, 2014).

### 2.1 Model averaging

For a continuous response it consists in averaging the predictions according to the following formula :

$$\hat{y} = \sum_{m=1}^{M} w_m f_m(x). \tag{1}$$

Bayesian model averaging (BMA) provides the following conceptual framework: Given a training set $T$, let $P(m/T)$ be the posterior probability of model $m$ given $T$. Then the posterior mean is $E(y/T) = \sum_{m=1}^{M} E(y/m,T)P(m/T) = \sum_{m=1}^{M} f_m(x)P(m/T)$. The bayesian prediction is a weighted average of the individual predictions with weights proportional to the posterior probability of each model (Hastie et al., 2009).

However, we will see in part 3 that formula (1) may also be used in a non Bayesian context.

### 2.2 Committee and ensemble methods

For a categorical response, *ie* a classification problem between $K$ classes, the majority rule consists in assigning an observation $x$ to the label $k$ which is the most frequent among the $M$ classifiers. This may be seen as a special case of model averaging with equal weights, hence the possibility of generalizing this method to unequal weights.

Committee methods are part of the vast literature on ensemble learning. For a comprehensive survey, see Zhou, 2012.

### 2.3 Local versus global learning

Model choice and model combinations stay the same for the entire space. If enough data are available, it becomes possible to find simultaneously a partition of the descriptor space into "competence regions" as Kuncheva (2014), and for each competence region $(c)$ a combination with specific weights $w_m^c$. It means that the first step consists in assigning a new observation $x$ to its competence region, and then to apply the adequate formula. This kind of approach is a generalization of clusterwise classification, itself an extension of clusterwise regression (Späth, 1979), but without the restriction of using only local linear models.

## 3 Stacking for a binary response

### 3.1 Stacked regressions

Originally developed by Wolpert (1992) in a machine learning context, stacking or stacked generalization is an ensemble method which has also been studied by Breiman

(1996) from a statistician's point of view. Let $f_m(x_i)$ be the prediction of a numerical response $y$ at point $x_i$ using some regression model $m$. Each model ($m = 1, ..., M$) may be of any kind: linear or non-linear, nonparametric, tree, neural nets etc. These base models are called *level-0* models. At *level-1*, stacking combines linearly the $M$ *level-0* predictors, with optimal weights $w_m$ according to a modified least squares criterion. It leads to a predictor which is better than any of a single *level-0* model $f_m(x)$.

A naive solution would consist in minimizing $\sum_{i=1}^{n}(y_i - \sum_{m=1}^{M} w_m f_m(x_i))^2$ directly on a training set: where $y_i$ is the observed response at point $x_i$.

However this is not a good idea since "we have not put each of the models on the same footing by taking into account their complexity" (Hastie et al., 2009). The more complex a model is, the higher will be its weight and over-fitting occurs.

Instead of standard predicted values, stacking uses $f_m^{-i}(x_i)$ the cross-validated prediction at $x_i$, not using $x_i$. This means that the weights minimize:

$$\sum_{i=1}^{n}\left(y_i - \sum_{m=1}^{M} w_m f_m^{-i}(x_i)\right)^2. \tag{2}$$

The final model is given by $\hat{y} = \sum_{m=1}^{M} w_m f_m(x)$.

If the weights are constrained to be positive and to sum to 1 (which is often recommended) stacking looks like a frequentist version of BMA. However unlike BMA, stacking does not need that all *level-0* models be of the same kind, nor that the true model belongs to the family. Experiments proved that stacking outperforms BMA in a large number of cases (Clarke, 2003) involving much simpler computations.

Leblanc and Tibshirani (1996) compared stacking with 3 other methods : least squares, bootstrap and Generalized Cross Validation (GCV). They conclude that stacking and bootstrap with non negativity constraints performed the best and that constraining the coefficients to have a sum to one was not efficient. In our opinion the fact that the non-negative estimators of the weights improves strongly stacking, is due to the high correlation between predictions of *level-0* models (multicollinearity) : in this case weights are unstable and may become negative. Imposing non-negativity realizes some regularization, but we will advocate another solution in part 3.3.

### 3.2 Stacked classifiers

The previous methodology may be adapted to supervised binary classification where the response variable is categorical: $y$ has two values 1 and 0 (or 1 and $-1$). *Level-0* models may be any kind of classifiers: Linear Discriminant Analysis (LDA), logistic regression, classification trees, Support Vector Machines (SVM) etc.

At *level-1*, the outcome being binary, the final combination of *level-0* predictors may be achieved by logistic regression, though Ting and Witten (1999) advocate the use of MLR, a linear machine which is a multi-response linear regression algorithm, which they compare to C4.5 and IB1, a modified $k$-nearest-neighbour technique. Surprisingly they did not compare MLR to the simple logistic regression which has the advantage of providing directly class probabilities, which we decided to use.

Note that the extension of stacking to generalized linear models at *level-1* is straightforward : $\hat{y} = \phi\left(\sum_{m=1}^{M} w_m f_m(x)\right)$ where $\phi$ is some link function. Though it is computationally feasible, it is not recommended to combine class predictions $\hat{y}_m$ where $\hat{y}_m$ takes values 1 and 0 (or 1 and $-1$) for the following reasons: class predictions depend on many choices like thresholds, prior class probabilities etc. Unbalanced categories for the response may lead to trivial solutions (100% predicted in one class). It is much more efficient to combine estimated membership probabilities $p_m(x)$ of class 1, in other terms fuzzy membership instead of crisp membership. In their comparative study, Ting and Witten (1999) demonstrated the effectiveness of stacked generalization. They found that the use of class probabilities was crucial for classification tasks rather than class predictions, but that non negativity constraints did not improve the performance in their examples. However non-negative weights are easier to interpret. See also Jacobs (1995).

### 3.3 Improving stacking methodology with PLS regression

We did not find any reference about the stability of stacking weights, though there is obviously a strong positive multicollinearity between *level-0* predictors as long as they are good predictors. This may be checked with a Principal Component Analysis (PCA) which usually shows a large first eigenvalue.

If we were in the regression case ($y$ numerical) we would to replace the cross-validated Ordinary Least Squares (OLS) estimation of the weights $w_m$ by a cross-validated regularized regression such as Principal Component Regression (PCR), ridge or PLS (Wold et al. (1983)). We advocate here the use of a single component PLS regression for its simplicity and the property that the weights will never be negative if all predictions $f_m(x)$ are positively correlated with $y$.

Since we deal with the classification case, PLS logistic regression as defined by Bastien et al. (2005) will be used with the R implementation PlsRglm (Bertrand et al., 2014). We made this choice which ensures to get probabilities instead of PLS-DA which may give a result out of the [0, 1] interval. Like for a numeric response, there is no need to impose non-negativity to the weights: they will be naturally positive as long as there is a positive link between the response and the probabilities $f_m(x)$.

## 4 Combining five families of models

Prediction methods for binary outcome belong to the wide set of supervised classification techniques. Among well-known statistical methods are Fisher's linear discriminant function and logistic regression, which have both proved their efficiency in many cases. However for complex phenomenon (e.g. biologics), these methods doesn't take easily into account some issues such as non-linearity, multicollinearity .... In order to counteract these problems, many other models have been developed as well by statisticians as by Machine Learning specialists such as: expert based scoring, decision trees, Bayesian networks, SVM.

Five prediction families have been retained here for *level-0* classifiers, covering a wide range of methods: linear models, classification trees, expert systems, nonlinear kernel machines, bayesian classifiers. Within each family we selected a specific technique: sparse PLS discriminant analysis among linear models, tree boosting among decision trees, an internally developed expert scoring, a radial based SVM among kernel machines and a modified naive Bayes classifier.

### 4.1 Sparse PLS discriminant analysis

Partial Least Squares Discriminant Analysis (PLS-DA) is derived from PLS regression for a categorical response in presence of multicollinearity; it has been proposed by Barker and Rayens (2003) and further studied by Noçairi et al. (2005).

Like PLS regression, PLS-DA is based on the iterative computation of "latent variables" which are linear combinations of the original descriptors. The first PLS-DA component is the solution of:

$$\max_{u,v} Cov(Xu, Yv), \tag{3}$$

where $X$ is the matrix of predictors and $Y$ the indicator matrix of the categories. $u$ and $v$ are the vectors of coefficients to be applied respectively to $X$ and $Y$. Higher order components are obtained by deflations of $X$ and $Y$ under orthogonality constraints.

The right number of components is obtaining by cross validation in order to get the best prediction of the response variable in terms of $R$-square.

When the response has two categories, PLS-DA provides a rather good linear classifier but which is a combination of all original predictors. It is an advantage when the number of predictors is low, but lead to uninterpretable results for high-dimensional data. Some selection becomes necessary. We have adapted to a categorical predictor context the Sparse-PLS regression proposed by (Chun and Keles, 2010), hence the name Sparse PLS-DA:

In the same spirit as the Lasso, (Tibshirani, 1996) S-PLS adds a $L1$ constraint to the regression coefficients which gives sparse loadings (ie many zero coefficients):

$$\max_{u} \left( u'X'YY'Xu \right) \; with \; ||u||^2 = 1 \; and \; \sum_{j=1}^{p} |u_j| < \lambda. \tag{4}$$

Chun and Keles (2010) reformulates the previous criterion by imposing $L_1$ penalty onto a surrogate direction vector $c$ instead of the original direction vector $u$, while keeping $u$ and $c$ close to each other:

$$\min_{\alpha,c} \left( -ku'X'YY'Xu + (1-k)(c-u)'X'YY'X(c-u) + \lambda_1||c||_1 + \lambda_2||c||^2 \right) \tag{5}$$

$$\text{with } u'u = c'c = 1.$$

The solution is obtained by alternatively iterating between solving for $c$ after fixing $u$ and solving for $u$ with fixed $c$ ; the initial value of $c$ being given by usual PLS. $k$ is an extraneous parameter controlling the local character of the solution.

### 4.2 Boosting

Boosting is a supervised classification method based on an approach developed by Shapire (1990) highlighting that combining weak learners may generate a single strong learner.

Several approaches were developed. Freund and Schapire (1997) proposed Adaboost, the first algorithm of boosting. This method allows increasing the quality of the prediction from a linear combination of simple but moderately precise rules.

Among the possible weak learners, Friedman et al. (2000) showed that those based on CART decision trees at two levels (Breiman, 1996) give good results. See also (Bühlmann and Hothorn, 2007).

Using boosting is interesting in case of small samples in comparison to decision trees, which needs more data as soon as levels increase in tree, in order to avoid a weak number of data in a terminal node.

For example, in case of 40 samples at 4 levels, with one balanced disjunction, we will only have 2 or 3 observations in the terminal nodes, while the boosting method based on decision trees at two levels, will have 10 observations in the terminal nodes (with one balanced disjunction).

Like classification trees, boosting allows nominal as well as continuous predictors and realizes a selection among them.

### 4.3 Support Vector Machines (SVM)

Support Vector Machines (Cortes and Vapnik, 1995) is widely used in machine learning for binary decision. This approach takes into account the fact that the predictors are potentially non-linearly related with the response variable. When data are linearly separable, the primary idea consists in finding the "thick" hyperplane which separates the data perfectly with a maximal margin (distance between the boundary and the closest observation).

When data are not separable by an hyperplane, they may be linearly separated after a transformation, which maps the data into an extended "feature space". An hyperplane in the feature space corresponds to a non-linear boundary in the input space. SVM uses the "kernel trick" which consists in defining the scalar product in the feature space by a transformation of the scalar product of the input space, which avoids computations in a high dimensional space. In this article, we use the gaussian kernel:

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2),$$

where $\gamma > 0$ is a regularization parameter which is usually optimized by cross-validation.

In theory, Support Vector Machines allow only continuous predictors. In case of a mixture of continuous and categorical predictors, the technique which consists in replacing each categorical predictor by the set of the indicator variables of its categories is generally not efficient. It is thus advised to perform a dimension reduction technique (multiple correspondence analysis) before applying the SVM methodology.

### 4.4 Naive Bayes classifier

Let $X_j$ be a binary predictor of a binary outcome and $p_0$ the frequency of the event $Y = 0$ (prior probability). Bayes formula relates the posterior probability $P(Y = 0/X_j = 0)$ to the sensitivity $Se_j$ and specificity $Sp_j$ of $X_j$.

$$P(Y = 0/X_j = 0) = \frac{p_0 P(X_j=0/Y=0)}{p_0 P(X_j=0/Y=0)+(1-p_0)P(X_j=0/Y=1)} = \frac{p_0 Se_j}{p_0 Se_j+(1-p_0)(1-Sp_j)}$$

Conversely, we have: $P(Y = 0/X_j = 1) = \frac{p_0(1-Se_j)}{p_0(1-Se_j)+(1-p_0)Sp_j}$

For $p$ predictors the naive Bayes classifier consists in computing the posterior probability $P(Y = 0/X_1, ..., X_p)$ by $\prod_{j=1}^{p} P(Y = 0/X_j)$ as if independence was true. A more appropriate term for this probabilistic model could be a "model for statistically independent features". Despite the fact that this model is generally wrong, the naive Bayes classifier performs often quite well (Hand and Yu, 2001).

Note that naive Bayes method allows only binary predictors. We select only variables with large enough specificity and sensitivity.

### 4.5 Expert Scoring

A specific score method has been developed for the needs of L'ORÉAL R&D (Gomes et al., 2014). Without going into details, the principle is the following: each predictor is converted into a partial score on a seven positions scale from -3 to +3, and the final score is the sum of partial scores. For instance, if a predictor is categorical, we assign a score equal to 3 to a category m if the ratio $n_{Am}/n_{Bm} > 3$ where $n_{Am}$ is the number of observations of the category of interest $A$ within category $m$, and $B$ is the complement of $A$. A symmetric rule $n_{Bm}/n_{Am} > 3$ gives a negative value -3. A similar rule is applied to numerical predictors which are also split into 7 ordered categories according to the overlapping of box-plots of $A$ and $B$ like for naive Bayes we select predictors with large enough specificity and sensitivity. This simple scoring technique has been found effective in a large number of cases and is well understood by users (dermatologists, biologists), though it is not based on a theoretical background.

### 4.6 Synthesis

Instead of choosing one particular technique, a meta model combining several of them (efficient and complementary in terms of performance) will lead to an improved decision rule.

Table 1: Properties of the 5 techniques

| Techniques | Ease of interpretation | Accept nonlinearity | Accept numerical predictors | Accept categorical predictors |
|---|---|---|---|---|
| Sparse PLS-DA | yes | no | yes | no |
| Boosting | no | yes | yes | yes |
| SVM | no | yes | yes | no |
| Naive Bayes | yes | yes | no | yes |
| Expert Scoring | yes | yes | yes | yes |

### 4.7 Example

In order to show the performances of this methodology, we used Heart Disease Data Set[1] ($n$=270), as an example. This database contains 76 attributes, but all published experiments refer to using a subset of $p$=14 of them. The heart disease score has been transformed into a binary outcome (0 = Absence; 1-4 = Presence of heart disease). All five models have been trained on $n$=189 units and evaluated on the same validation set ($n$=81). As expected the five probabilities of presence are highly correlated, see table 2; thus PLS regularization was fully justified. The stacking weights of the five models

Table 2: Correlations between level-0 outputs

| **Models** | **Boosting** | **SVM** | **Expert Scoring** | **Sparse PLSDA** | **Naive Bayes** | **Stacking** |
|---|---|---|---|---|---|---|
| **Boosting** | 1 | | | | | |
| **SVM** | 0.92 | 1 | | | | |
| **Expert Scoring** | 0.86 | 0.89 | 1 | | | |
| **Sparse PLS-DA** | 0.89 | 0.97 | 0.88 | 1 | | |
| **Naive Bayes** | 0.88 | 0.93 | 0.90 | 0.94 | 1 | |
| **Stacking** | 0.92 | 0.98 | 0.92 | 0.98 | 0.97 | 1 |

are given by Table 3. They are positive, without having imposed non-negativity as a

[1]Heart Disease Data Set: Bache, K. & Lichman, M. (2013). -UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

constraint.

Table 3: stacking weights

| **Models** | $w^j$ |
|---|---|
| **Boosting** | 1.8031 |
| **SVM** | 1.4561 |
| **Expert Scoring** | 1.8522 |
| **Sparse PLS-DA** | 1.2225 |
| **Naive Bayes** | 1.2321 |

Table 4: Performance comparisons on the validation set ($n$=81). Best one in bold.

| Performances | Boosting | SVM | Expert Scoring | Sparse PLS-DA | Naive Bayes | Stacking all models | Stacking Score SPLS-DA |
|---|---|---|---|---|---|---|---|
| **True positives** | 29 | 29 | **30** | 29 | 29 | 29 | **30** |
| **False positives** | 12 | 8 | 8 | **7** | 11 | **7** | 6 |
| **False negatives** | 7 | 7 | **6** | 7 | 7 | 7 | 6 |
| **True negatives** | 33 | 37 | 37 | 38 | 34 | 38 | **39** |
| **Sensitivity** | 80.6 | 80.5 | **83.3** | 80.6 | 80.6 | 80.5 | **83.3** |
| **Specificity** | 73.3 | 82.2 | 82.2 | 84.4 | 75.6 | 84.4 | **86.6** |
| **Concordance** | 76.5 | 81.5 | 82.7 | 82.7 | 77.8 | 82.7 | **85.2** |
| **Kappa** | 0.53 | 0.62 | 0.65 | 0.65 | 0.55 | 0.65 | **0.7** |
| **AUC** | 0.862 | **0.909** | 0.896 | 0.880 | 0.884 | **0.909** | **0.909** |

The best models on the training set are not necessarily the best models on the validation set. The stacking based on the five models in this example is not better than the best of the five sub-models but has a better specificity.

Here, we can improve the result by selecting only two models instead of five : less complexity, better generalization. In terms of Cohen's kappa, the two best models are the Score and the Sparse PLS DA (kappa = 0.65) which are complementary: Score model is the best one in terms of sensitivity, and sparse PLS-DA is the best one for specificity.

Indeed the stacked model based only on combining score and sparse PLS DA shows higher performances than all the models taken separately. It provides the highest response rate (85.2% of validation set) and the best balance between sensitivity (83.3%) and specificity (86.6%).

# 5 A heuristic learning process in case of small data set

In our specific applications we were faced to two critical issues: the weak number of observations and the presence of categorical data. For a small sample, the choice of the learning set may bring some bias in the choice of pertinent variables, thus it is necessary to perform repeated sampling. Regarding the second issue, it is necessary to avoid during each random sub-sampling that some categories become empty. We present the implementation of the solutions.

## 5.1 Robust variable selection

According to a standard process, we create two sub samples (step 1): learning L and validation V.

In learning set L, each model is calibrated (step 3) by cross-validation (for example, figure 1 shows in details the calibration process for the boosting model) and provides its own choice of variables. Since boosting, sparse PLS-DA, Naive Bayes and Expert Scoring provide four possibly different selection of variable. However this choice of variables is not robust because it may strongly depend on the sample L drawn.

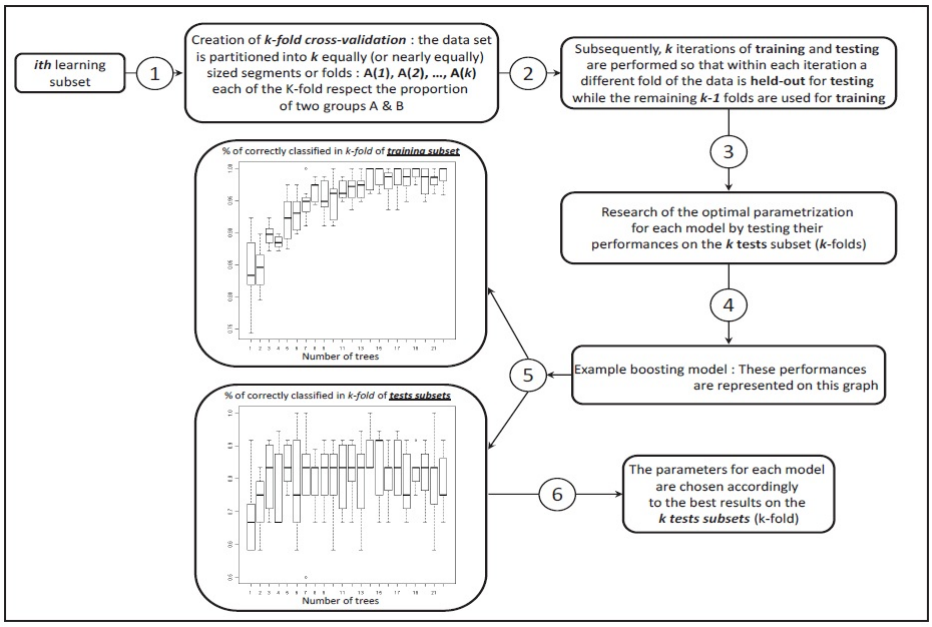



Figure 1: Model parametrization process

So, in order to avoid any bias induced by this choice, we proceed (Step 2) in the following way: we draw **s** subsets of $L$ to obtain a hard core of variables common to

different sampling and the four methods. We keep at the end the predictors which are selected more than 3**s** times.

The final model $F$ is estimated on the sample $L$ (step 4) with all the common variables retained from the **s** samples, and its performance is measured on $V$.

The following figure represents the process used to build a meta-model and compute its performances on a learning/test split data set (see figure 2).

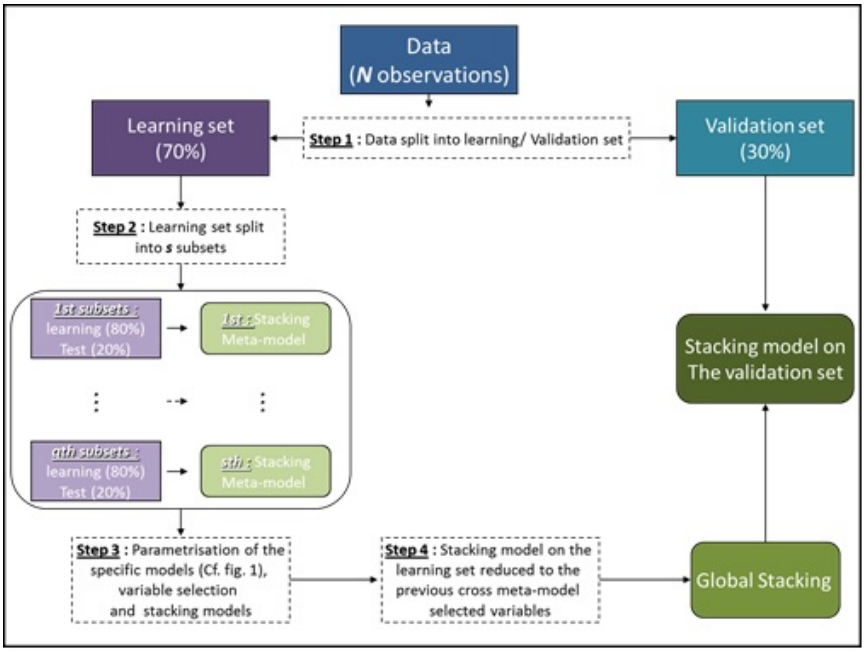



Figure 2: Process of validation rules

## 5.2 Construction of the learning and test samples in case of categorical predictors

In the case of a small data set, sampling into a learning and a test subsets may lead to some empty categories of both the response (rarely) and (frequently) of the predictors. In this case it is impossible to estimate some parameters. In any case, it is suitable to have a minimum number of observations in each category of each variable.

Our solution relies on a specific stratification technique:

First a balanced stratification according to the categories of the outcome $y$ is necessary to keep constant their proportions. But some categories of the predictors may become empty or with not enough observations. A theoretical solution would consist in a fully balanced sampling scheme (Deville and Tillé (2004)) but is not feasible for a small data set.

The following heuristics is then used:

- Perform a random split into learning and test data sets with a stratified sampling

upon both categories of the outcome

- Reject a sample if a category of a predictor has not enough representatives in both the learning and the test samples
- Repeat until acceptance
- Repeat until getting 6 balanced samples.

# 6 Application to cosmetics tolerance data

## 6.1 Statutory context

The 7th Amendment of the European Cosmetic Directive has banned the *in vivo* tests on animals for the safety evaluation of ingredients. L'ORÉAL has thus developed several types of *in vitro*, *in silico* methods and collected other kinds of information on its chemicals like physico-chemical data. Due to the complexity of the skin sensitization (or irritation) process, it is now agreed that it is necessary to use all these informations to predict safety.

We will focus on a specific end-point: the skin sensitization i.e, if a chemical is a sensitizer or not (danger/non danger). The statistical objective in this case is to predict the *in vivo* tests results realized before their ban, by using *in vitro* and *in silico* data.

The data set was composed of $N$=165 chemicals characterized by $p$=35 variables, representing the results from *in silico* predictions, *in vitro* tests, assays as well as numerous physico-chemical experimental or calculated parameters. The list of these variables being industry confidential will not be detailed here.

The prediction model is based on the stacking methodology described in the previous parts of this paper and summarized in figure 3. In addition, we propose a decision system based on the construction of intervals ("traffic light" zones "red", "green", and "orange" for no decision) in order to have a robust conclusion.

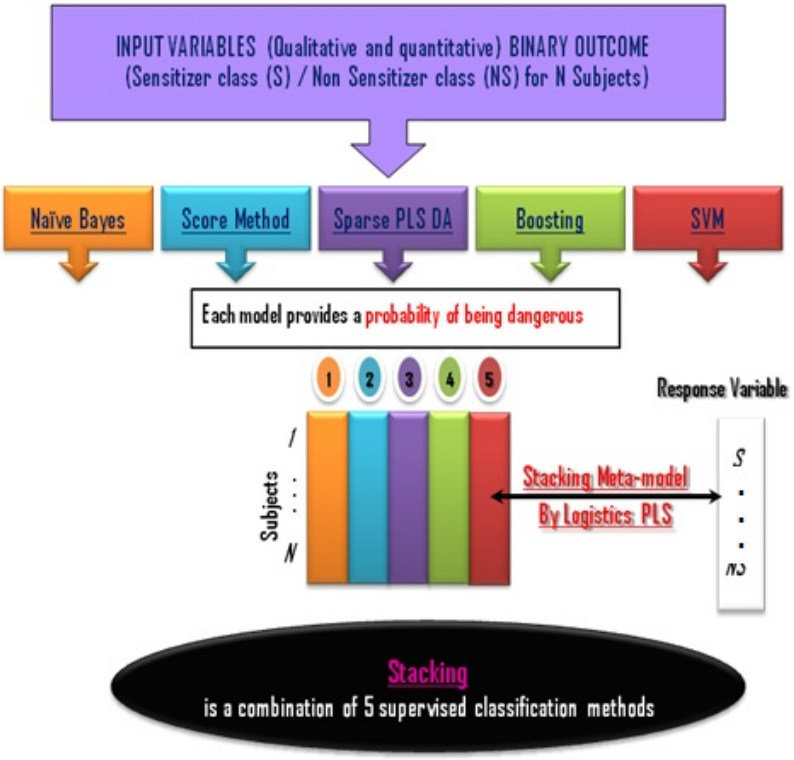



Figure 3: Visualization of the methodology

## 7 The Klimisch score

In the context of cosmetics tolerance data, we use a modified version of the naive Bayes classifier in order to take into account a prior information on the reliability of each test $X_j$ known as the Klimisch score (Klimisch et al., 1997). Klimisch score is a method of assessing the reliability of toxicological studies, mainly for regulatory purposes.

Based on the Klimisch score, we define a quality factor QF which is used to correct the sensitivity and specificity of each test as follows (Gomes et al., 2014):

Klimisch score 1: "reliable result" $\rightarrow$ QF =1
Klimisch score 2: "doubtful result" $\rightarrow$ QF =0.8
Klimisch score 3: "unreliable result" $\rightarrow$ QF =0.2
If the data is missing then QF is equal to 0.
Corrected sensitivity = 0.5 + QF * (Sensitivity - 0.5)
Corrected specificity = 0.5 + QF * (Specificity - 0.5)

### 7.1 Results

Each model, including stacking, provides a probability to be dangerous.

In this safety application, the variable selection procedure of part 5 retains only 10 variables, among the 35.

As expected, the predictions provided by the five models are clearly highly positively correlated, see tables 5, 6 figure 4 and 5:

Table 5: Correlations between predicted probabilities

| **Models** | **Boosting** | **SVM** | **Expert Scoring** | **Sparse PLS-DA** | **Naive Bayes** |
|---|---|---|---|---|---|
| **Boosting** | 1 | | | | |
| **SVM** | 0.88 | 1 | | | |
| **Expert Scoring** | 0.88 | 0.92 | 1 | | |
| **Sparse PLS-DA** | 0.87 | 0.99 | 0.91 | 1 | |
| **Naive Bayes** | 0.85 | 0.89 | 0.93 | 0.87 | 1 |

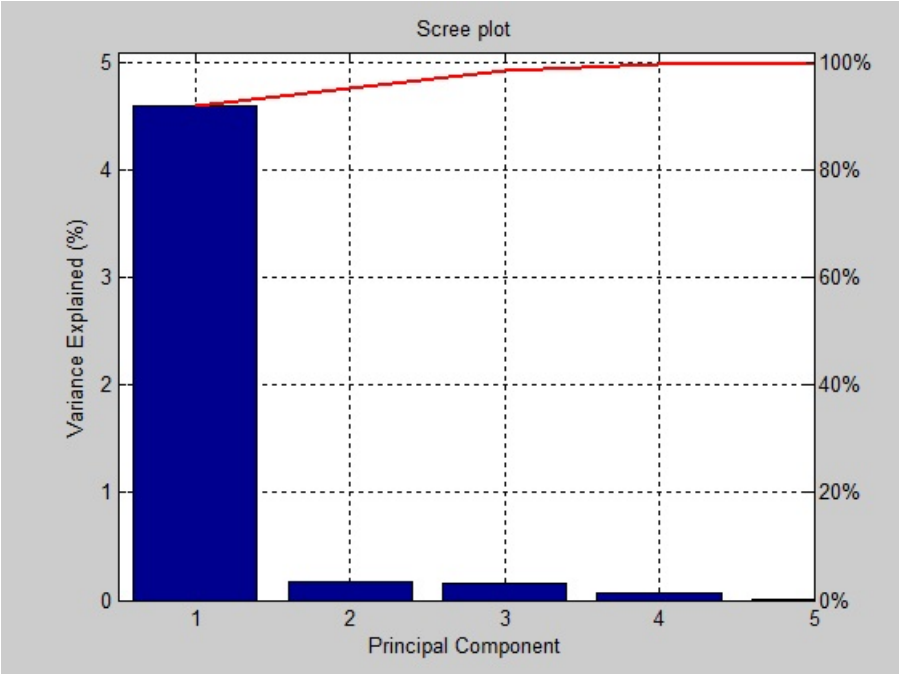



Figure 4: Eigenvalues by components

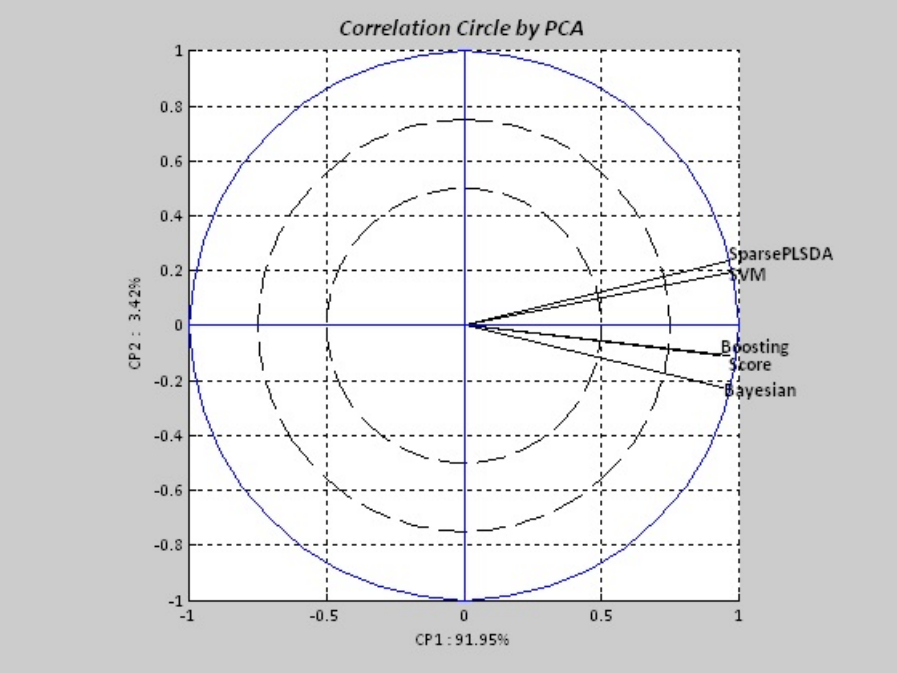



Figure 5: Correlations with principal components

The stacking meta-model has been computed by Logistic PLS-DA: optimal weights are given in table 6. Once again, we note that weights are positive. Figure 6 presents ROC curves of the five models plus the stacking meta-model on the learning set. Stacking appears to be the most efficient (blue curve) with the highest area under the curve (0.949).

Table 6: stacking weights

| **Models** | $w^j$ |
|---|---|
| **Boosting** | 1.656 |
| **SVM** | 1.609 |
| **Expert Scoring** | 2.281 |
| **Sparse PLS-DA** | 1.311 |
| **Naive Bayes** | 1.188 |

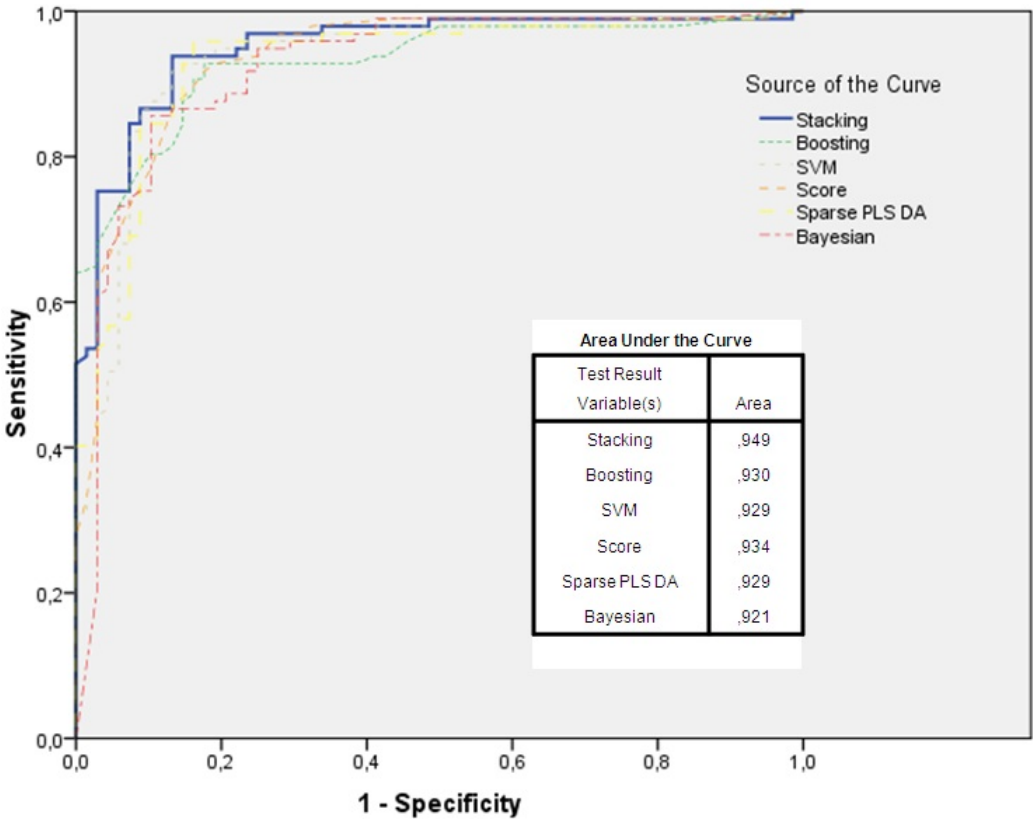



Figure 6: ROC curves for the training set

As described in the methodology in figure 1, we split data into training and validation sets ($N = 115$ of data will be for training and $N = 50$ to use for validation sets). Then, we repeated this partition 10 times. We have evaluated the performances on the 10 validation sets.

Table 7 shows that the stacking model provides the highest response rate (90% average of 10 validation set) and the best balance between average sensitivity (89.7%) and average specificity (95.94%). However naive Bayes performs quite well.

Table 7: Performance comparisons on the validation set (N=50)

| Performances | Boosting | SVM | Expert Scoring | Sparse PLSDA | Naive Bayes | Stacking |
|---|---|---|---|---|---|---|
| Nbr-Concl. | 23.5 ± 2.2 | 37.3 ± 1.8 | 22.4 ± 1.9 | 37.8 ± 1.6 | 40.8 ± 1.5 | **44.9 ± 1.4** |
| TP | 13.1 ± 2.1 | 19.4 ± 1.0 | 13.1 ± 2.0 | 19.3 ± 0.9 | 21.7 ± 1.2 | **23.3 ± 0.7** |
| FP | 0.0 ± 0.0 | 1 ± 0.9 | 0.2 ± 0.3 | 1.0 ± 0.9 | 0.9 ± 0.4 | 1 ± 0.9 |
| FN | 0.8 ± 0.5 | 1.5 ± 0.38 | 0.2 ± 0.3 | 1.5 ± 0.4 | 1.5 ± 0.4 | 2.7 ± 0.5 |
| TN | 9.6 ± 1.4 | 15.3 ± 1.0 | 8.9 ± 1.9 | 16.0 ± 1.2 | 16.7 ± 0.8 | **17.9 ± 0.8** |
| Sensitivity | 94.1 ± 3.3 | 92.8 ± 1.9 | 98.6 ± 2.1 | 92.7 ± 1.9 | 93.5 ± 1.4 | 89.7 ± 1.4 |
| Specificity | **100** ± 0.0 | 94.2 ± 4.6 | 97.8 ± 3.3 | 94.2 ± 4.9 | 94.9 ± 2.3 | 95.9 ± 0.6 |
| Concordance | 96.53 ± 2.0 | 93.4 ± 1.7 | **98.2 ± 2.2** | 93.5 ± 1.7 | 94.1 ± 1.1 | 92.0 ± 1.2 |
| Kappa | 0.93 ± 0.23 | 0.87 ± 0.24 | 0.94 ± 0.34 | 0.86 ± 0.26 | 0.86 ± 0.23 | 0.86 ± 0.15 |
| AUC | 0.87 ± 0.01 | 0.89 ± 0.02 | 0.87 ± 0.02 | 0.86 ± 0.02 | 0.85 ± 0.02 | **0.92** ± 0.01 |

Another way to assess the efficiency of stacking is the following:

- Instead of giving a unique threshold for the danger probability (eg 0.5), we use a partition with two thresholds into 3 intervals corresponding respectively to "danger", "no danger" and "no decision" (unconclusive) defined as follows:
  - Chemicals with a probability >= 85% are predicted Danger,
  - Chemicals with a probability <= 15% are predicted Non Danger,
  - Chemicals with a probability between those two thresholds are inconclusive.

This approach defines a kind of "reliability area". For example for a one learning sample and validation, Boosting is conclusive on only 40% (67/165) of chemicals while stacking is conclusive on 82% (135/165) of chemicals as shown in figure 7. This reliability area, or in other words the "response rate", is an important indicator in order to evaluate the stacking performance. Here stacking leads to a conclusion over more chemicals.

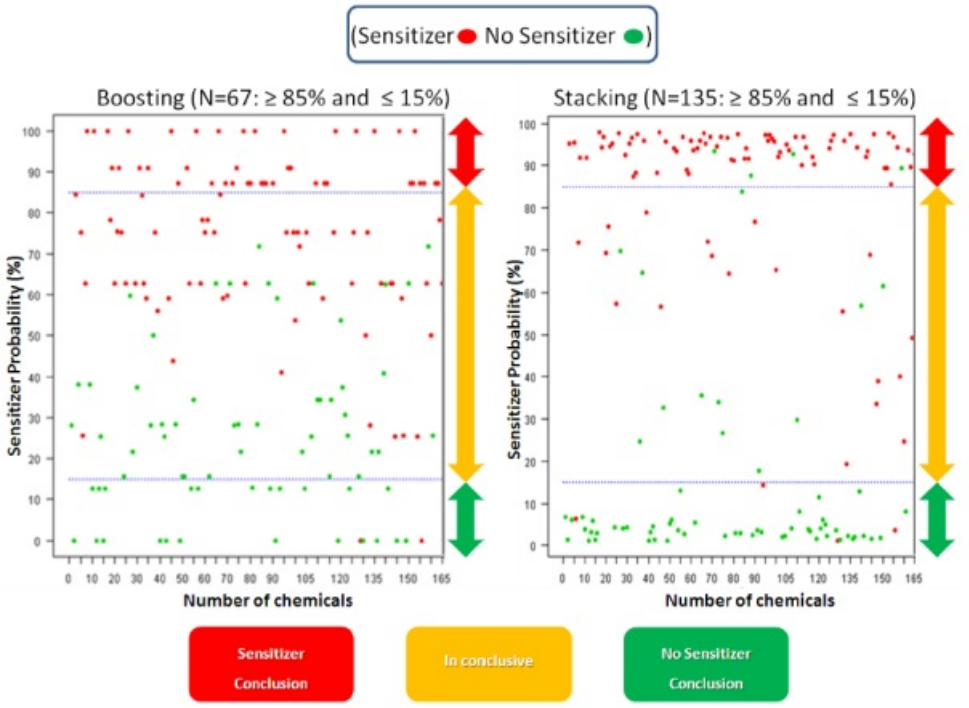



Figure 7: Reliability areas

Furthermore, we observe that the distribution of the "danger" probabilities provided by stacking look more bimodal than all other models, see Figure 8 which compares probabilities provided by stacking and boosting.

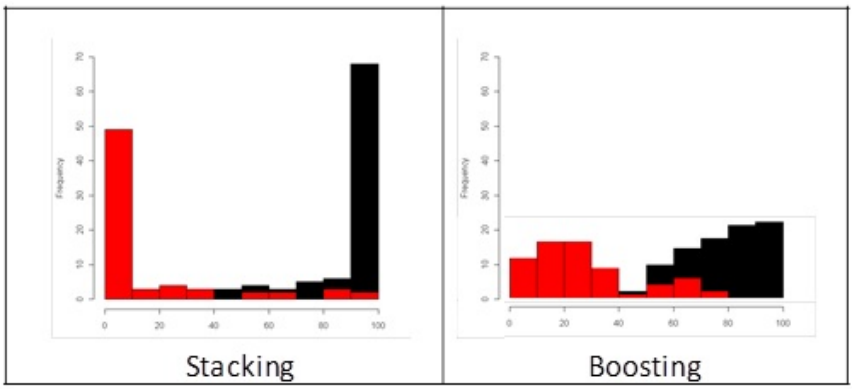



Figure 8: Danger probabilities

## 8 Conclusion and perspectives

We have presented some modifications to stacking methodology for a binary outcome, taking into account the correlations between predictions.

We have proposed an heuristic methodology based on sample stratification to overcome the problem of a too small number of units, which makes difficult to split the data into learning/test subset.

By combining five known classifiers of very different kinds, we obtained a prediction model with better performances than each of the five initial models taken separately. This result is important for the development of alternative approaches in safety evaluation of chemicals in cosmetic industry.

Various extensions are possible : adding other level-0 models such as Decision Trees like C4.5 (Quinlan, 1993), Neural Networks (Anderson, 1995), Multiblock Redundancy Analysis (Bougeard et al., 2006), partitioning around medoids (PAM: Kaufman and Rousseeuw, 1987) ...

Extensions to the multi-class response case are in progress, either for ordered or non ordered categories.

**Nota:** All computations have been done in the R environment combined with a web interface where most of these methods were already available. We used the following R packages: pls , spls , rpart , plsRglm , adabag , e1071 , penalizedSVM. Figure 5 was realized with IBM-SPSS 19.

## Acknowledgement

The authors are very grateful to Cécile Piroird, Silvia Teissier, Thierry Pauloin, Nathalie Aleppee, Valérie Michaut, José Cotovio and Fréderic Leroy for fruitful discussions and careful and critical reading of the manuscript.

## References

Anderson, J.A. (1995). *An Introduction to Neural Networks. Cambridge.* MA: MIT Press.

Barker, M. and Rayens, W. (2003). PLS for discrimination. *J. Chemometrics*, 17 : 166-173.

Bastien, P., Esposito-Vinzi, V. and Tenenhaus, M. (2005). PLS generalised linear regression. *Computational Statistics & Data Analysis*, 48 : 17-46.

Bertrand, F., Magnanensi, J., Meyer, N. and Maumy-Bertrand, M. (2014). plsRglm: Algorithmic insights and applications. Edited: June 2014; Compiled: July 17, 2014 https://cran.r-project.org/web/packages/plsRglm/plsRglm.pdf

Bougeard, S., Hanafi, M., Noçairi, H., and Qannari, E.M. (2006). Multiblock canonical correlation analysis for categorical variables: application to epidemiological data. *In Greenacre, M., Blasius, J. Eds Multiple correspondence analysis and related methods. Chapman & Hall/CRC*, 393-404.

Breiman, L. (1996); Stacked regressions. *Machine Learning*, 24 : 49-64.

Bühlmann, P., and Hothorn, T. (2007). Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science*, 22, 7, 477-505.

Chun, H. and Keles, S. (2010). Sparse partial least squares for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society*, Series B , Vol. 72, p. 3-25.

Clarke, B. (2003); Comparing Bayes Model Averaging and Stacking When Model Approximation. *Journal of Machine Learning Research*, 4, 683-712.

Cortes, C. and Vapnik, V. (1995). Support-vector network. *Machine Learning*, 20, 1-25.

Deville, J.C. and Tillé ,Y. (2004). Efficient balanced sampling : The cube method. *Biometrika*, 91 (4) : 893-912.

Freund, Y. and Schapire, R. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55, 1, 119-139.

Friedman, J. H., Hastie, T. and Tibshirani, R. (2000). Additive Logistic Regression: a Statistical View of Boosting. *The Annals of Statistics*, Vol. 28, No. 2, 337-407.

Gomes, C., Noçairi , H. , Thomas, M. , Ibanez, F. , Collin, J. , Saporta, G. (2012). Stacking prediction for a binary outcome. *Proceedings of Compstat 2012, 20th International Conference on Computational Statistics, Limassol, Cyprus*, 271-282.

Gomes, C., Noçairi , H. , Thomas, M., Collin, J. , Saporta, G. (2014). A simple and robust scoring technique for binary classification. *Artificial Intelligence Research*, vol. 3(1), 52-58.

Hand, D.J. and Yu, K. (2001). Idiot's Bayes-not so stupid after all? *International Statistical Review*, 69, 385-398.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning.* 2nd edition, Springer.

Jacobs, R.A. (1995). Methods for combining experts' probability assessments. *Neural computation*, 7, 867-888.

Kaufman, L. and Rousseeuw, P.J. (1987). Clustering by means of medoids (PAM). In Dodge,Y. (ed.). *Statistical Data Analysis Based on the L1-norm and Related Methods. North Holland, Amsterdam*, 405-416.

Klimisch, H.J. , Andreae, M. and Tillmann, U. (1997). A Systematic Approach for Evaluating the Quality of Experimental Toxicological and Ecotoxicological Data. *Regulatory Toxicology and Pharmacology*,25, 1-5.

Kuncheva, L. (2014). *Combining pattern classifiers, methods and algorithms.* 2nd edition, J.Wiley & sons.

Leblanc, M. and Tibshirani, R. (1996). Combining Estimates in Regression and Classification. *Journal of the American Statistical Association*, 91, 436, 1641-1650.

Noçairi, H., Qannari, E.M., Vigneau E., and Bertrand D. (2005). Discrimination on latent components with respect to patterns. Application to multicollinear data. *Computational Statistics & Data Analysis*, 48, 139-147.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Shapire, R.E. (1990). The strength of weak learnability. *Machine Learning*, 5(2),197-227.

Späth, H. (1979). Clusterwise linear regression. *Computing*, 22, 367-373.

Sesmero, Paz. Ledezma, M., and Sanchis, A. (2015). Generating ensembles of heterogeneous classifiers using stacked generalization. *WIREs Data Mining Knowl. Discov.*, 5, 21-34.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser.*, B 58, 267-288.

Ting, K.M. and Witten, I.H. (1999). Issues in stacked generalization. *J. Artif. Intell. Res*, 10, 271-289.

Wold S., Martens, H. and Wold, H. (1983). The multivariate calibration problem in chemistry solved by the PLS method. *In : Matrix pencils. Springer Berlin Heidelberg*, 286-293.

Wolpert, D. (1992). Stacked Generalization. *Neural Networks*, 5, 41-259

Zhou, Z-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall.