Benchmarking random forest:

a large scale experiment

Raphael Couronné[1], Philipp Probst[1], Anne-Laure Boulesteix[1]

[1] Department of Medical Informatics, Biometry and Epidemiology, University of Munich (LMU), Marchioninistr. 15, D-81377 Munich, Germany.

Abstract

1 Introduction

In the low dimensional world, logistic regression is considered a standard approach to binary classification. This is especially true in scientific fields such as medicine or psychosocial sciences where the focus is not only on prediction but also on explanation; see Schmuheli (Statistical Science 2010) for a discussion of this distinction. Since their invention 15 years ago, random forests Breiman (2001) have strongly gained in popularity and are increasingly becoming a common "standard tool" used by scientists without any strong background in statistics or machine learning. Our experience as authors, reviewers and readers is that random forest can now be used routinely without the audience strongly questioning this choice. While their use was in the early years limited to innovation-friendly scientists interested (or experts) in machine learning, it has now become commonplace. Random forests are well-known in various non-computational communities.
In this context, we think that the performance of the method should be systematically investigated in a large-scale benchmarking experiment and compared to the current standard: logistic regression. We make the—admittedly somewhat controversial—choice to consider the standard version of RF only, with default parameters, and logistic regression only as the standard approach which is very often considered in a first step for low dimensional binary classification problem. We also investigate the dependence of our conclusions on datasets' characteristics. In particular, as a important by-product of our study, we provide insights into the importance of inclusion criteria for datasets in benchmarking experiments and more generally critically discuss design issues and scientific practice in this context. This paper is structured as follows. After a short overview of LR and RF, as well as the associated VIM and partial dependance plots Friedman (2001), we present the methodology of the benchmark, including the criteria for the dataset's selection. The paper then emphasizes of the analysis of the performance in regards with the dataset's characteristics.

2 Methods

2.1 Logistic regression (LR)

2.1.1 Model

A population is divided into two classes, represented with the vector $\boldsymbol{Y}$ with $Y_i \in \{0, 1\}$, $i \in \{1, ..., p\}$. Each observation is described with p features, $\boldsymbol{X} = (\boldsymbol{X_1}, ..., \boldsymbol{X_p})$. The logistic regression model estimates the probability of a binary response using the logistic function on a linear function from X. We use the following notations : π_i = $\mathbb{P}(Y = 1|\{X_{i,1}, ..., X_{i,p}\})$, and $\boldsymbol{\beta} = \{\beta_0, ..., \beta_p\}^T$ the parameter to be estimated with maximum likelihood. The regression logistic model is defined with the following formula :

$$logit(\pi_i) = ln\frac{\pi_i}{(1-\pi_i)} = \beta_0 + \sum_{j=1}^{p} \beta_j * X_{i,j}. \tag{1}$$

2.2 Random forest (RF)

2.2.1 Brief overview

The random forest is an "ensemble learning" technique which extends the model of decision tree by aggregating a large number of decision trees, and averaging their result, resulting in a lower variance. We present here the random forest first described in Breiman (2001). Each tree of the forest is built using a bootstrap sample of the original dataset, using the CART method and the Decrease Gini Impurtiy (DGI) as the splitting criterion. During the building of each tree of the forest, at each split, only $mtry$ variables are considered (different subset at each split). Note random Forest is considered a black-box algorithm, as gaining insight on a RF model is hard due to the huge number of trees. Some methods specific to the random forest exist to gain information, probably the most important being the VIMs presented in section 2.2.3. Another problem is the transportability of the random forest, as no convention exists on the implementation of the algorithm.

2.2.2 Parameters

We present here the most important parameters for the random forest, and their common default value (as presented in package $randomForest$). As we aim to evaluate the performance of the standard random forest on out dataset, we will use these default values. Note that hyperparameter tuning is the subject of ongoing research, **for more details see Philip Probst**. $ntree$ denotes he number of trees in the forest. Theorically, increasing the number of trees always yields more reliable results. Its value should be high enough so that each candidate predictor has enough opportunities to be selected. Default value is 500 in package $randomForest$. Then $mtry$ denotes the number of candidate predictors randomly considered at each split. A low value gives more opportunities to predictors with small effects which may contribute to accurate prediction. A high value reduces the risk of having only non-informative predictors at hand. Default value is $\sqrt{p}$ for classification and $\frac{p}{3}$ for regression. $nodesize$ represents the minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown. Default value is 1 for classification and 5 for regression. $replace$ refers to the resampling scheme chosen to obtain the differents subsamples on which the trees are grown. In general, the method is a bootstrap sample with or without replacement, default value is $True$.

2.2.3 Variable importance measures

Random Forest can be used to rank internally the importance of the variables (Breiman, 2001). The random forest computes two difference variable importance : the Gini VIM and the permutation VIM. The Gini importance corresponds to the sum of the DGI for all the splits of the forest corresponding to the chosen variable divided by the number of trees, it reflects the importance of the variable in the construction of the tree using DGI. The permutation VIM is based on the accuracy. For a variable, compute the difference of the OOB error before and after permuting randomly the values of the considered variable. An important predictor is expected to reduce more accuracy when neutralized with the permutation. Note that the Gini VIM has bias, for example with the number of candidates

for the split, which issue is adressed in Strobl et al. (2007).

VIs are not sufficient to capture the patterns of dependency between predictors and response. They only indicate—in the form of a single number—whether there is such a dependency. Partial dependence plots can be used to address this shortcoming. They can essentially be applied to any prediction method but are particularly useful for black-box methods which (in contrast to, say, generalized linear models) do not yield any interpretable patterns.

2.3 Partial dependence plots

2.3.1 Principle

Partial dependence plots (PDPs) offer an insight of any black box machine learning model, visualizing how each feature influence the prediction by averaging the prediction on all the other features. The PDPs method was developed first introduced by Friedman (2001) for his gradient boosting machine. Let F denote the mathematical function associated with the model, j the index of the chose feature X_j and $\boldsymbol{X}_{\bar{j}}$ the complement subset such that $\boldsymbol{X}_{\bar{j}} = \{\boldsymbol{X}_1, ..., \boldsymbol{X}_{j-1}, \boldsymbol{X}_{j+1}, ..., \boldsymbol{X}_n\}$. The partial dependance of F on $\boldsymbol{X}_j$ is

$$F_{X_j} = \mathbb{E}_{X_{\bar{j}}} F(X_j, X_{\bar{j}}) \tag{2}$$

which can be estimated from the data using the empirical distribution

$$\hat{F}_{X_j}(x) = \frac{1}{N} \sum_{i=1}^{N} F(x_{i,1}, ...x_{i,j-1}, x, x_{i,j+1}, ..., x_{i,p}). \tag{3}$$

As an illustration, we display in figure 1 the partial dependence plots obtained by logistic regression and random forest for three simulated datasets of size $n = 1000$. The datasets are simulated according to the formula $logit(\mathbb{P}(Y = 1)) = \beta_0 + \beta_1 x_1 + \beta_2 x_1 x_2 + \beta 3 x_1^2$. The first dataset (on the top) represents the linear scenario ($\beta_2 = \beta 3 = 0$), the second dataset an interaction ($\beta 1 = \beta 3 = 0$) and the third a cases of non-linearity ($\beta 1 = \beta 2 = 0$). For all three

datasets the random vector $(X_1, X_2)^\top$ follows the distribution $\mathcal{N}_2(0, I)$, with I representing the identity matrix.

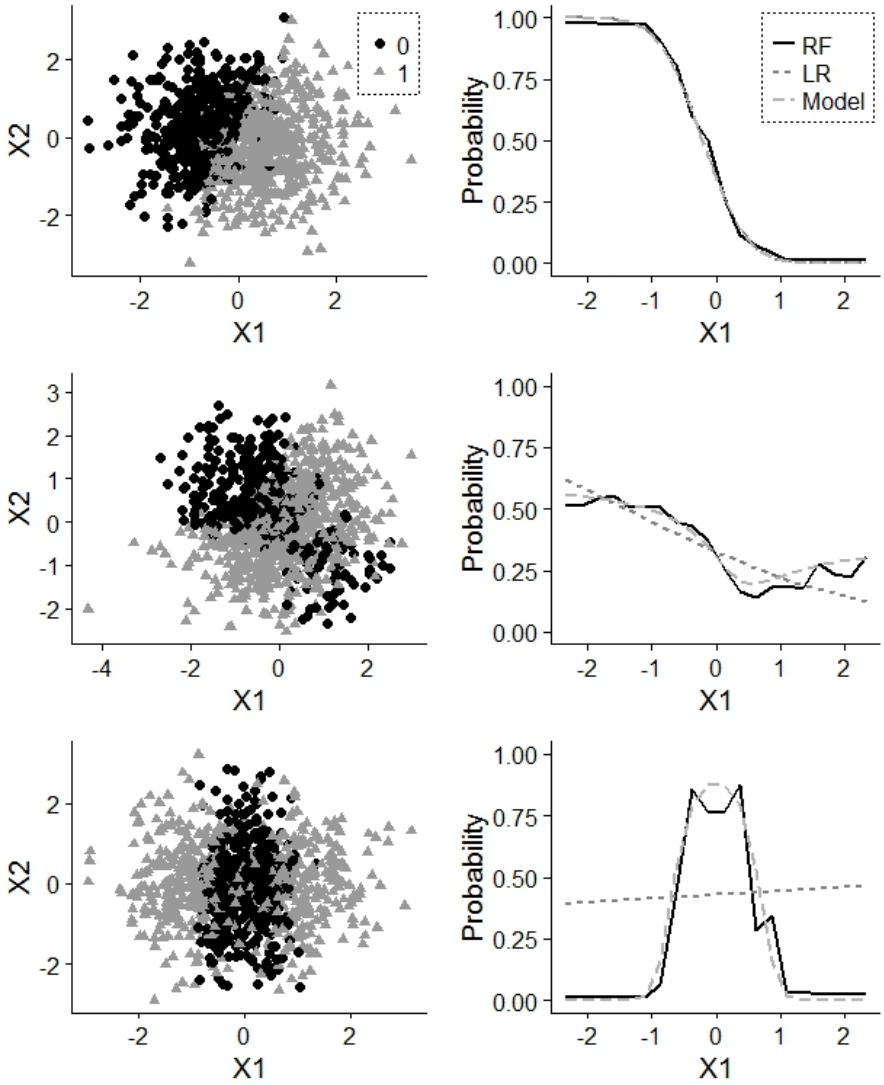

Figure 1: Plot of the PDP for the tree simulated datasets. Each line is related to a dataset. On the left, visualization of the dataset. On the right, the partial dependance for the variable X_1. The datasets represent from top to bottom a linear, dependant, and non linear relation with the target.

2.4 Benchmarking with real data

In this section we present the design of our benchmarking experiment. Most importantly, the experiment is based on a collection of M real datasets (in contrast to other types of benchmarking experiments relying on simulated data). The prediction accuracy of LR and RF on real datasets is estimated through cross-validation as briefly presented in Section 2.4.1. Issues related to the statistical analysis of the benchmarking results as discussed in Boulesteix et al. (The American Statistician 2015) are reviewed in Section 2.4.3.

2.4.1 Cross-validation

In a k-fold cross-validation, the original sample is randomly split in k subsamples of equal sizes. One of the fold is chosen as the test set, the k-1 others are used for training. The process is repeated k times, and then the performances are averaged. We chose a 10 times 5-Cross Validation in our benchmark (the 5-CV is repeated 10 times), see Bischl et al. (2012) for advice on resampling methods. In the stratified version of the CV, the folds are chosen such that we have the same proportion of the classes in all the partitions. The stratified version was chosen because the proportion of imbalanced dataset was important. With too imbalanced datasets, it may happen that a level of a categorical feature is present in the test set, but not present in the train set, in which case the logistic regression cannot be applied. At the end we obtain results in the form of a $M \times 2$ data matrix containing the CV errors of LR (first column) and RF (second column) for the M considered datasets.

2.4.2 Accuracy measures

Given a classifier, let $\hat{f}(i, j)$ represent the estimated probability of observation i belonging to class j, $f(i, j) = 1_{ij}$ the true probability of example i to be of class j. $C(i, j) \in \{0, 1\}$, it equals 1 if $f(i, j)$ made the right prediction, else 0. n is the number of observations of the dataset, and c the number of classes. In our study, we consider the following measures quantifying prediction accuracy in the case of a binary classification problem (Ferri et al., 2009).

- The Accuracy, proportion of correct prediction:
  $Acc = \sum_{i=1}^{N} \sum_{j=1}^{c} f(i,j)C(i,j)$

- The Area Under Curve, average of the error for each class:
  $Auc = \frac{\sum_{i=1}^{N} f(i,j) \sum_{t=1}^{n}}{n_j n_k} I(f(i,j)f(t,j))$

- The Brier Score, which penalizes strong deviations from the true probability:
  $Brier = \frac{\sum_{j=1}^{c} \sum_{i=1}^{n} (f(i,j) - \hat{f}(i,j))^2}{n}$

The time of training was also added.

2.4.3 Statistical analysis

summary of Boulesteix et al. (The American Statistician 2015) **???**

2.4.4 The OpenML database

So far we have said that the benchmarking experiment used a collection of M real datasets without specifying which ones. In practice, one often uses already formatted datasets from public databases for this purpose. Many of them offer a user-friendly interface and a good documentation which facilitate to some extent the preliminary steps of the benchmarking experiment (search for datasets, data download, preprocessing). One of the most well-known such databases is UCI repository, see Lichman (2013). Specific scientific areas may have their own databases, such as ArrayExpress and GEO for molecular data from high-throughput experiments, see Brazma et al. (2003) and NCBI (2013) respectively. Most recently, the OpenML database (Vanschoren et al., 2014) has been initiated as an exchange platform allowing machine learning scientists to share their data and results. This database includes as many as 19625 datasets as of october 2016, a non-negligible proportion of which are relevant as example datasets for benchmarking classification methods.

2.4.5 Inclusion criteria

When using a huge database of datasets, it becomes obvious that one has to define criteria for inclusion in the benchmarking experiment. Inclusion criteria in this context do not have any long tradition in computational science. The criteria used by researchers to select datasets are most often completely non-transparent. It is often the fact that they select a number of datasets which were found to somehow fit the scope of the investigated methods, but without clear definition of this scope.

We conjecture that datasets are occasionally removed from the experiment *a posteriori* because the results do not meet the expectations/hopes of the researcher. While the vast majority of researchers certainly do not cheat consciously, such practices may substantially bias the conclusion of a benchmarking experiment; see for instance Yousefi et al. (2010) **is it the right one ? Yousefi et al. (Bioinformatics 2010)** for theoretical and empirical investigation of this problem. In a word, "fishing for datasets" should be prohibited, see Rule 4 of Boulesteix (2015).

Even if fishing for datasets is prohibited, it is important that criteria for inclusion in the benchmarking experiment are clearly stated; **see Boulesteix, Wilson and Hapfelmeier (technical report 2016, coming soon)** for an extensive discussion of this issue.

In our study, we consider the following datasets' characteristics to define inclusion criteria:

- n: number of observations
- p: number of features
- $\frac{p}{n}$
- d : dimension of the dataset
- $\frac{d}{n}$
- C_{min} Percentage of elements of the minority class

- C_{max} Percentage of elements of the majority class
- $p_{numeric}$: number of numeric features
- $p_{categorical}$: number of categorical features
- $timetrain$: duration for the run a 5-fold CV with a default Random Forest

Based on these datasets' characteristics, we define several sets of inclusion criteria and investigate the impact of this choice on the results of the benchmarking experiment. In the same vein, one can also analyse the results of benchmarking experiments for different subsets of datasets successively, following the principle of subgroup analyses performed in clinical trials. For example, one could analyse the results for "large" datasets ($n > 1000$) and "small datasets" ($n \leq 1000$) separately.

2.4.6 Meta-learning

Going one step further, one can try to model the difference between the methods' performances based on the datasets' characteristics. Such a modelling approach can be seen as a simple form of *meta-learning*—a well-known task in machine learning.

3 Results

We consider a set of M datasets . Each one of its observations is a dataset from openML. For each observation we compute the performance of the different learners we consider: random forest, logistic regression and its penalized versions. Several packages are used: mlr for higher abstraction and a simpler way to compute benchmarks (Bischl et al., 2016), openML for loading the datasets (Casalicchio et al., 2016), and snowfall for parallel computing (R Core Team, 2016). We give more details on the parameters of the benchmark in the subsections.

3.1 Datasets

Details on the criteria for dataset selection are important. On the 20000 datasets from OpenML, we select the binary classification problem. We remove the datasets that include missing values, the high dimentional datasets such that $p > n$, and the obvious simulated datasets to consider 240 datasets.

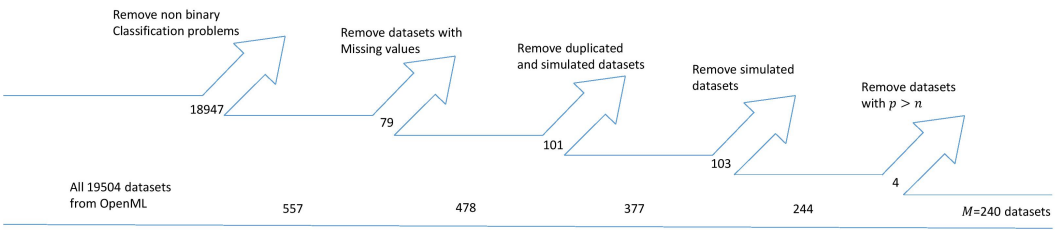

Figure 2: Flowchart representing the criteria for selection of the datasets.

Distribution of the datasets characteristics ?

3.2 Overall results

	acc	auc	$brier$	ber	$logloss$	$timetrain$
Logistic regression	0.825	0.830	0.129	0.235	1.169	0.152
Random forest	0.853	0.869	0.103	0.211	0.374	3.133
	acc	auc	$brier$	ber	$logloss$	$timetrain$
Logistic regression	1.710	1.720	1.720	1.614	1.684	1.008
Random forest	1.290	1.280	1.280	1.386	1.316	1.992

Table 1: Table representing the performances for both LR and RF. On the top, the mean performance values. On the bottom, the mean ranks.

Overall performances are presented in table 1 for all the measures. For the following sections, we decide to focus on the accuracy, as correlation is high between the different performance measures, and the results similar. We observe in figure 3 the boxplots of performances of Random Forest and Logistic Regression. We also plotted the boxplot of the difference in accuracy, and observed that Random Forest does better in most of the cases (71 % of our datasets), and when logistic regression outperforms random forest the difference is minimal. The Nemeyi's test

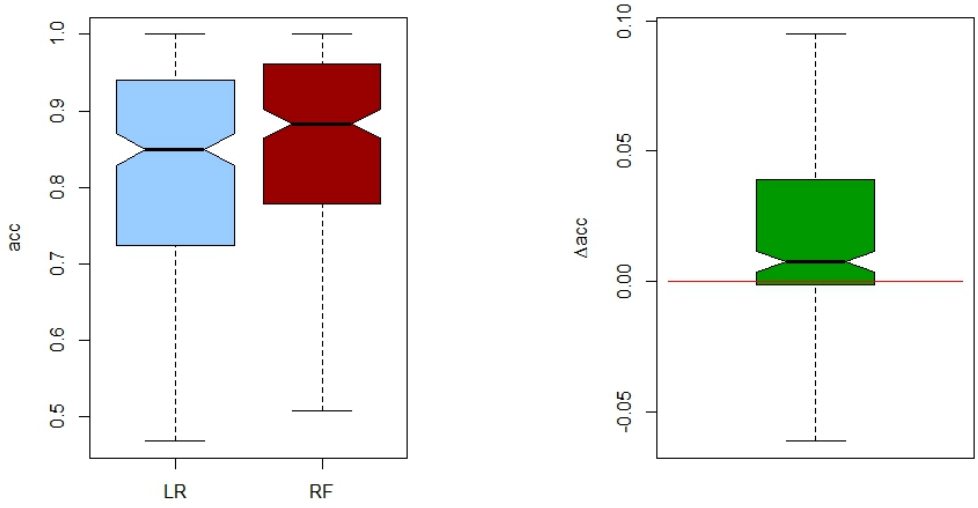

Figure 3: On the left, boxplot of the performance in accuracy for the random forest (in red) and the logistic regression (in blue). On the right, boxplot of the difference in accuracy between random forest and logistic regression. A red line is plotted at y=0.

3.3 Explaining differences: datasets' characteristics

3.3.1 Principle

While it is obvious to any computational scientist that the performance of methods may depend on some datasets' characteristics, this issue is not easy to investigate in real data settings because i) it requires a large number of datasets—a condition that is often not fulfilled in practice; ii) this problem is enhanced by the correlations between characteristics. In our benchmarking experiment, however, we consider such a huge number of datasets that an investigation of the relationship between methods' performances and datasets' characteristic becomes possible to some extent. We decide to illustrate this idea using only one dataset from OpenML in figure 4 with the datasets's feature p^*. For different values of p, $p \in \{1, p^*\}$ we randomly choose N times p features from our dataset to feed our algorithms and evaluate the performance. We observe in figure 4 that the accuracy increases for both logistic regression and random forest, as we expected. We also see that the accuracy of random forest increases faster than for logistic regression. Thus p seems like an important criteria when choosing between a random forest and a logistic regression.

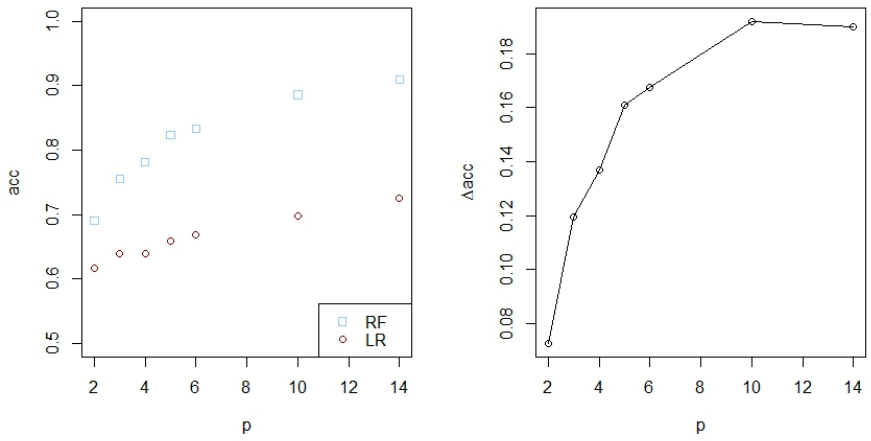

Figure 4: On the left, mean evaluation of the performance of Rf and LR on the OpenML dataset (Id = 1496) for different values of p. On the left, plot of the difference of the mean performance $\Delta acc = acc_{RF} - acc_{LR}$. Here $p \in \{2, 4, 5, 6, 10, 14\}$ and the performance is evaluated with a 2 times 5-CV.

3.3.2 Results with varying inclusion criteria

We decided to extend this idea, and studied the influence of the dataset's parameter on the difference in accuracy between random forest and logistic regression (Δacc). We computed the p-values of the Kendall and Spearman's test, and the significance value for the associated linear model. Results are shown in table 2. It seems the most relevant dataset's features are p and its derivatives and $Brier_{LR}$.

	Spearman's ρ p-value	Kendall's τ p-value	$\mathbb{P}(>\|t\|)$
n	$3.52e^{-1}$	$4.11e^{-1}$	$1.55e^{-1}$
p	$1.35e^{-6}$	$6.67e^{-7}$	$3.10e^{-3}$
$\frac{p}{n}$	$7.01e^{-2}$	$9.67e^{-2}$	$8.9e^{-1}$
d	$5.50e^{-4}$	$3.95e^{-4}$	$4.30e^{-3}$
$\frac{d}{n}$	$8.74e^{-2}$	$1.42e^{-3}$	$8.25e^{-1}$
$p_{numeric}$	$2.78e{-}4$	$1.56e{-}4$	$2.89e{-}1$
$p_{categorical}$	$1.73e{-}1$	$1.64e{-}1$	$4.10e{-}3$
$p_{numeric,rate}$	$5.87e{-}4$	$2.96e{-}4$	$6.27{-}1$
$p_{categorical,rate}$	$5.87e{-}4$	$2.96e{-}4$	$6.27{-}1$
C_{min}	$7.73e{-}1$	$6.98{-}1$	$6.87{-}2$
C_{max}	$7.73e{-}1$	$6.98{-}1$	$6.87{-}2$
$Brier_{LR}$	$5.35e{-}4$	$3.16{-}4$	$4.64{-}8$

Table 2: Table representing the correlation between Δacc and the corresponding dataset's features.

We decide to

3.3.3 Subgroup analysis as additional file? ?? I don't understand this part ??

3.3.4 Meta-learning

Do a Cart Tree and see ??? How much performance ?

3.4 Explaining differences: partial dependence plots

- In the previous section we have investigated the impact of datasets' characteristics on the results of benchmarking and simply modeled the difference between methods' performance based on these characteristics.

- In this section, we take a different approach to the explanation of differences. We use partial dependence plots as a technique to assess the dependency pattern between response and predictors underlying the prediction rule. When the methods' performances are different, we intuitively expect these dependency patterns to be different, sampling variations put aside. We typically expect this to be the case when the true joint distribution of response and predictors is far from the logistic regression model or when the logistic regression model holds but RF fails to recover it (due to, e.g., too small sample size).

-

4 Test Bibliography

References

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Jones, Z., Casalicchio, G., 2016. mlr: Machine Learning in R. R package version 2.10.
URL https://github.com/mlr-org/mlr

Bischl, B., Mersmann, O., Trautmann, H., Weihs, C., 2012. Resampling methods for meta-model validation with recommendations for evolutionary computation. Evolutionary Computation 20 (2), 249–275.
URL http://www.mitpressjournals.org/doi/pdf/10.1162/EVCO_a_00069

Boulesteix, A.-L., 2015. Ten simple rules for reducing overoptimistic reporting in methodological computational research. PLoS Comput Biol 11 (4), e1004191.

Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G. G., et al., 2003. ArrayexpressâĂŤa public repository for microarray gene expression data at the ebi. Nucleic acids research 31 (1), 68–71.

Breiman, L., 2001. Random forests. Machine learning 45 (1), 5–32.

Casalicchio, G., Bischl, B., Kirchhoff, D., Lang, M., Hofner, B., Bossek, J., Kerschke, P., Vanschoren, J., 2016. OpenML: Exploring Machine Learning Better, Together. R package version 1.0.
URL https://github.com/openml/openml-r

Ferri, C., Hernández-Orallo, J., Modroiu, R., 2009. An experimental comparison of performance measures for classification. Pattern Recognition Letters 30 (1), 27–38.

Friedman, J. H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189–1232.

Lichman, M., 2013. UCI machine learning repository.
URL http://archive.ics.uci.edu/ml

NCBI, G., 2013. archive for functional genomics data sets-update barrett.

R Core Team, 2016. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL https://www.R-project.org/

Strobl, C., Boulesteix, A.-L., Zeileis, A., Hothorn, T., 2007. Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC bioinformatics 8 (1), 1.

Vanschoren, J., Van Rijn, J. N., Bischl, B., Torgo, L., 2014. Openml: networked science in machine learning. ACM SIGKDD Explorations Newsletter 15 (2), 49–60.

Yousefi, M. R., Hua, J., Sima, C., Dougherty, E. R., 2010. Reporting bias when using real data sets to analyze classification performance. Bioinformatics 26 (1), 68–76.

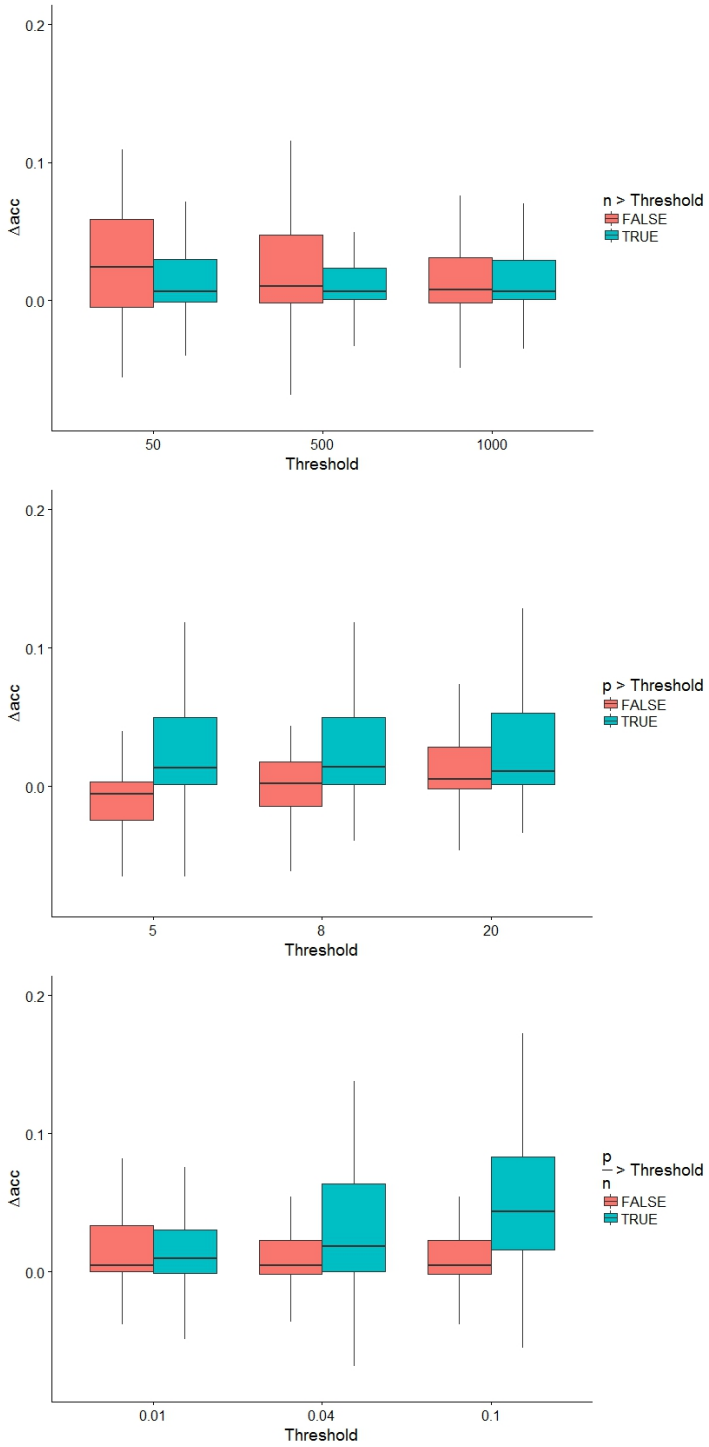

Figure 5: Boxplots of Δacc for different threshold as criteria for dataset's selection. On the top, boxplots for feature n, on the middle for feature p and on the bottom for the feature $\frac{p}{n}$.