LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Raphael Couronné, Philipp Probst, Anne-Laure Boulesteix

# Random forest versus logistic regression: a large-scale benchmark experiment

Technical Report Number 205, 2017
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

# Random forest versus logistic regression: a large-scale benchmark experiment

Raphael Couronne*[1], Philipp Probst†[1], Anne-Laure Boulesteix‡[1]

[1]Department of Medical Informatics, Biometry and Epidemiology, LMU Munich, Germany

July 19, 2017

**Abstract**

The Random Forest (RF) algorithm for regression and classification has considerably gained popularity since its introduction in 2001. Meanwhile, it has grown to a standard classification approach competing with logistic regression in many innovation-friendly scientific fields. In this context, we present a large scale benchmarking experiment based on 260 real datasets comparing the prediction performance of the original version of RF with default parameters and LR as binary classification tools. Most importantly, the design of our benchmark experiment is inspired from clinical trial methodology, thus avoiding common pitfalls and major sources of biases.

RF performed better than LR according to the considered accuracy measured in approximately 69% of the datasets. The mean difference between RF and LR was 0.032 (95%-CI=$[0.025, 0.042]$) for the accuracy, 0.043 (95%-CI=$[0.032, 0.056]$) for the Area Under the Curve, and $-0.028$ (95%-CI=$[-0.036, -0.022]$) for the Brier score, all measures thus suggesting a significantly better performance of RF. As a side-result of our benchmarking experiment, we observed that the results were highly dependent on the inclusion criteria used to select the example datasets, thus emphasizing the importance of clear statements regarding this dataset selection process. We also stress that neutral studies similar to ours, based on a high number of datasets and carefully designed, will be necessary in the future to evaluate further variants, implementations or parameters of random forests which may yield improved accuracy compared to the original version with default values.

*raphael.couronne@gmail.com

†probst@ibe.med.uni-muenchen.de

‡boulesteix@ibe.med.uni-muenchen.de

# 1 Introduction

In the context of low-dimensional data (i.e. when the number of covariates is small compared to the sample size), logistic regression is considered a standard approach for binary classification. This is especially true in scientific fields such as medicine or psycho-social sciences where the focus is not only on prediction but also on explanation; see Shmueli [1] for a discussion of this distinction. Since its invention 16 years ago, the random forest (RF) prediction algorithm [2], which focuses on prediction rather than explanation, has strongly gained popularity and is increasingly becoming a common "standard tool" also 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 in many scientific fields without particular justification and without the audience strongly questioning this choice. While its use was in the early years limited to innovation-friendly scientists interested (or experts) in machine learning, random forests are now well-known in various non-computational communities.

In this context, we believe that the performance of RF should be systematically investigated in a large-scale benchmarking experiment and compared to the current standard: logistic regression (LR). We make the—admittedly somewhat controversial—choice to consider the standard version of RF only with default parameters — as implemented in the widely used R package `randomForest` [3] version 4.6-12 — and logistic regression only as the standard approach which is very often used for low dimensional binary classification.

The rationale behind this simplifying choice is that, to become a "standard method" that users with different (possibly non-computational) backgrounds select by default, a method should be simple to use and not require any complex human intervention (such as parameter tuning) demanding particular expertise. Our experience from statistical consulting is that applied research practitioners tend to apply methods in their simplest form for different reasons including lack of time, lack of expertise and the (critical) requirement of many applied journals to keep data analysis as simple as possible. Currently, the simplest approach consists of running RF with default parameter values, since no unified and easy-to-use tuning approach has yet established itself. It is not the goal of this paper to discuss how to improve RF's performance by appropriate tuning strategies and which level of expertise is ideally required to use RF. We simply acknowledge that the standard variant with default values is widely used and conjecture that things will probably not dramatically change in the short term. That is why we made the choice to consider RF with default values as implemented in the very widely used package `randomForest`—while admitting that, if time and competence are available, more sophisticated strategies may often be preferable.

Comparison studies published in literature often include a large number of methods but a relatively small number of datasets [4], yielding an ill-posed problem as far as statistical interpretation of benchmarking results are concerned. In the present paper we take an opposite approach: we focus on only two methods for the reasons outlined above but design our benchmarking experiments

in such a way that it yields solid evidence. A particular strength of our study is that we as authors are equally familiar with both methods. Moreover, we are "neutral" in the sense that we have no personal *priori* preference for one of the methods: ALB published a number of papers on RF, but also papers on regression-based approaches [5, 6] and papers pointing to critical problems of RF [7, 8, 9]. Neutrality and equal expertise would be much more difficult if not impossible to ensure if several variants of RF (including tuning strategies) and logistic regression were included in the study. Further discussions of the concept of authors' neutrality can be found elsewhere [4, 10].

Most importantly, the design of our benchmark experiment is inspired by the methodology of clinical trials that has been developed with huge efforts for several decades. We follow the line taken in our recent paper [10] and carefully define the design of our benchmark experiments including, beyond issues related to neutrality outlined above, considerations on sample size (i.e. number of datasets included in the experiment) and strict inclusion criteria for datasets. Moreover, as an analogon to subgroup analyses and the search for biomarkers of treatment effect in clinical trials, we also investigate the dependence of our conclusions on datasets' characteristics.

As an important by-product of our study, we provide empirical insights into the importance of inclusion criteria for datasets in benchmarking experiments and general but critical discussions on design issues and scientific practice in this context. The goal of our paper is thus two-fold. Firstly we aim to present solid evidence on the performance of standard logistic regression and random forests with default values. Secondly, we demonstrate the design of a benchmark experiment inspired from clinical trial methodology.

The rest of this paper is structured as follows. After a short overview of LR and RF, the associated VIM, partial dependance plots [11], the cross-validation procedure and performance measures used to evaluate the methods (Section 2), we present our benchmarking approach in Section 3, including the criteria for dataset selection. Results are presented in Section 4.

# 2 Background

This section gives a short overview of the (existing) methods involved in our benchmarking experiments: logistic regression (LR), random forest (RF) including variable important measures, partial dependence plots, and performance evaluation by cross-validation using different performance measures.

## 2.1 Logistic regression (LR)

Let $Y$ denote the binary response variable of interest and $X_1, \ldots, X_p$ the random variables considered as explaining variables, termed *features* in this paper. The logistic regression model links the conditional probability $P(Y = 1|X_1, ..., X_p)$

to $X_1, \ldots, X_p$ through

$$P(Y = 1|X_1, ..., X_p) = \frac{\exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)}{1 + \exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)}, \quad (1)$$

where $\beta_0, \beta_1, \ldots, \beta_p$ are regression coefficients, which are estimated by maximum-likelihood from the considered dataset. The probability that $Y = 1$ for a new instance is then estimated by replacing the $\beta$'s by their estimated counterparts and the $X$'s by their realizations for the considered new instance in Eq. (1). The new instance is then assigned to class $Y = 1$ if $P(Y = 1) > c$, where $c$ is a fixed threshold, and to class $Y = 0$ otherwise. The commonly used threshold $c = 0.5$, which is also used in our study, yields a so-called Bayes classifier.

## 2.2 Random forest (RF)

### 2.2.1 Brief overview

The random forest (RF) is an "ensemble learning" technique consisting of the aggregation of a large number of decision trees, resulting in a reduction of variance compared to the single decision trees. In this paper we consider Leo Breiman's original version of RF [2], while acknowledging that other variants exist, for example RF based on conditional inference trees [12] which address the problem of variable selection bias [13] and perform better in some cases, or extremely randomized trees [14].

In the original version of RF [2], each tree of the RF is built based on a bootstrap sample drawn randomly from the original dataset using the CART method and the Decrease Gini Impuritiy (DGI) as the splitting criterion [2]. When building each tree, at each split, only a given number `mtry` of randomly selected features are considered as candidates for splitting. RF is usually considered a black-box algorithm, as gaining insight on a RF prediction rule is hard due to the large number of trees. One of the most common approaches to extract from the random forest interpretable information on the contribution of different variables consists in the computation of the so-called variable importance measures outlined in Section 2.2.3. In this study we use the package `randomForest` [3] (version 4.6-12) with default values, see the next paragraph for more details on tuning parameters.

### 2.2.2 Hyperparameters

This section presents the most important parameters for RF and their common default values as implemented in the R package `randomForest` [3] and considered in our study. Note, however, that alternative choices may yield better performance [15, 16] and that parameter tuning for RF has to be further addressed in future research. The parameter `ntree` denotes the number of trees in the forest, which should be in principle as large as possible so that each candidate feature has enough opportunities to be selected. The default value is `ntree`=500 in the package `randomForest`. The parameter `mtry` denotes the

number of features randomly selected as candidate features at each split. A low value increases the chance of selection of features with small effects, which may contribute to improved prediction performance in cases where they would otherwise be masked by features with large effects. A high value of `mtry` reduces the risk of having only non-informative candidate features. In the package `randomForest`, the default value is $\sqrt{p}$ for classification with $p$ the number of features of the dataset. The parameter `nodesize` represents the minimum size of terminal nodes. Setting this number larger causes smaller trees to grow. The default value is 1 for classification. The parameter `replace` refers to the resampling scheme used to randomly draw from the original dataset different samples on which the trees are grown. The default is `replace=TRUE`, yielding bootstrap samples, as opposed to `replace=FALSE` yielding subsamples.

### 2.2.3 Variable importance measures

As a byproduct of random forests, the built-in variable importance measures (VIM) rank the *variables* (i.e. the features) with respect to their relevance for prediction [2]. The so-called Gini VIM has shown to be strongly biased [13]. The second common VIM, called permutation-based VIM, is directly based on the accuracy of RF: it is computed as the mean difference (over the `ntree` trees) between the OOB errors before and after randomly permuting the values of the considered variable. The underlying idea is that the permutation of an important feature is expected to decrease accuracy more strongly than the permutation of an unimportant variable.

VIMs are not sufficient in capturing the patterns of dependency between features and response. They only reflect—in the form of a single number—the strength of this 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

Partial dependence plots (PDPs) offer insight of any black box machine learning model, visualizing how each feature influences the prediction while averaging with respect to all the other features. The PDP method was first developed for gradient boosting [11]. Let $F$ denote the function associated with the classification rule: for classification, $F(X_1, \ldots, X_p) \in [0, 1]$ is the predicted probability of the observation belonging to class 1. Let $j$ be the index of the chosen feature $X_j$ and $X_{\bar{j}}$ its complement, such that $X_{\bar{j}} = \{X_1, ..., X_{j-1}, X_{j+1}, ..., X_p\}$. The partial dependance of $F$ on feature $X_j$ is the expectation

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

which can be estimated from the data using the empirical distribution

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

where $x_{i,1}, \ldots, x_{i,p}$ stand for the observed values of $X_1, \ldots, X_p$ for the $i$th observation. As an illustration, we display in Figure 1 the partial dependence plots obtained by logistic regression and random forest for three simulated datasets representing classification problems, each including $n = 1000$ independent observations. For each dataset the variable $Y$ is simulated according to the formula $\log(P(Y = 1)/P(Y = 0)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2$. The first dataset (top) represents the linear scenario ($\beta_1 \neq 0$, $\beta_2 \neq 0$, $\beta_3 = \beta_4 = 0$), the second dataset (middle) an interaction ($\beta_1 \neq 0$, $\beta_2 \neq 0$, $\beta_3 \neq 0$, $\beta_4 = 0$) and the third (bottom) a case of non-linearity ($\beta_1 = \beta_2 = \beta_3 = 0$, $\beta_4 \neq 0$). For all three datasets the random vector $(X_1, X_2)^\top$ follows distribution $\mathcal{N}_2(0, I)$, with $I$ representing the identity matrix. The data points are represented in the left column, while the PDPs are displayed in the right column for RF, logistic regression as well as the true logistic regression model (i.e. with the true coefficient values instead of fitted values). We see that RF captures the dependance and non-linearity structures in cases 2 and 3, while logistic regression, as expected, is not able to.

## 2.4 Performance assessment

### 2.4.1 Cross-validation

In a $k$-fold cross-validation (CV), the original dataset is randomly partitioned into $k$ subsets of approximately equal sizes. At each of the $k$ CV iterations, one of the folds is chosen as the test set, while the $k - 1$ others are used for training. The considered performance metric is computed based on the test set. After the $k$ iterations, the performances are finally averaged over the iterations. In our study, we perform 10 repetitions of stratified 5-fold CV, as commonly recommended [17]. In the stratified version of the CV, the folds are chosen such that the class frequencies are approximately the same in all folds. The stratified version is chosen mainly to avoid problems with strongly imbalanced datasets occurring when all observations of a rare class are included in the same fold. By "10 repetitions", we mean that the whole CV procedure is repeated for 10 random partitions into $k$ folds with the aim to provide more stable estimates.

In our study, this procedure is applied to different performance metrics outlined in the next subsection, for LR and RF successively and for $M$ real datasets successively. For each performance metric, the results are stored in form of an $M \times 2$ matrix.

### 2.4.2 Performance measures

Given a classifier and a test dataset of size $n_{test}$, let $\hat{p}_i$, $i = 1, \ldots, n$ denote the estimated probability of the $i$th observation ($i = 1, \ldots, n_{test}$) to belong to class

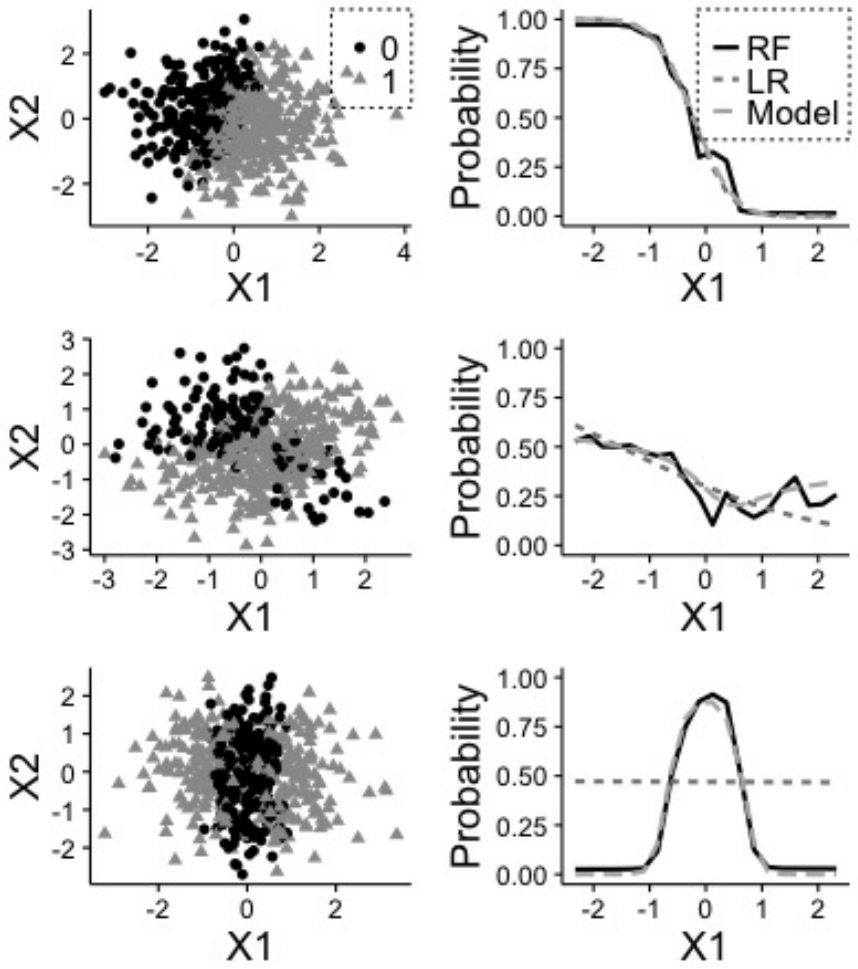


Figure 1: Exemple of partial dependence plots
Plot of the PDP for the three simulated datasets. Each line is related to a dataset. On the left, visualization of the dataset. On the right, the partial dependance for the variable $X_1$. First dataset: $\beta_0 = 1, \beta_1 = 5, \beta_2 = -2$ (linear), second dataset: $\beta_0 = 1, \beta_1 = 1, \beta_2 = -1, \beta_3 = 3$ (interaction), third dataset $\beta_0 = -2, \beta_4 = 5$ (non-linear).

$Y = 1$, while the true class membership of observation $i$ is simply denoted as $y_i$. Following the Bayes rule implicitly adopted in LR and RF, the predicted class $\hat{y}_i$ is simply defined as $\hat{y}_i = 1$ if $\hat{p}_i > 0.5$ and 0 otherwise.

The *accuracy*, or proportion of correct predictions is estimated as

$$Acc = \frac{1}{n_{test}} \sum_{i=1}^{n_test} I(y_i = \hat{y}_i),$$

where $I(.)$ denotes the indicator function ($I(A) = 1$ if $A$ holds, $I(A) = 0$ otherwise). The *Area Under Curve* (auc), or probability that the classifier ranks a randomly chosen observation with $Y = 1$ higher than a randomly chosen observation with $Y = 0$ is estimated as

$$auc = \frac{1}{n_{0,test} n_{1,test}} \sum_{i:y_i=1} \sum_{j:y_j=0} I(\hat{p}_i > \hat{p}_j),$$

where $n_{0,test}$ and $n_{1,test}$ are the numbers of observations in the test set with $y_i = 0$ and $y_i = 1$, respectively. The *Brier Score*, measuring the deviation between true class and predicted probability, is estimated as

$$Brier = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (\hat{p}_i - y_i)^2.$$

In addition to these three measures of prediction performance, we also consider the training computation time as an additional criterion.

# 3 Benchmarking approach

## 3.1 The OpenML database

So far we have stated that the benchmarking experiment uses a collection of $M$ real datasets without further specifications. In practice, one often uses already formatted datasets from public databases. Some of these databases offer a user-friendly interface and 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 database is the UCI repository [18]. Specific scientific areas may have their own databases, such as ArrayExpress for molecular data from high-throughput experiments [19]. More recently, the OpenML database [20] 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.

## 3.2 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 does not have any long tradition in computational science. The

criteria used by researchers—including ourselves before the present study—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, from published studies, datasets are occasionally removed from the experiment *a posteriori* because the results do not meet the expectations/hopes of the researchers. While the vast majority of researchers certainly do not cheat consciously, such practices may substantially introduce bias to the conclusion of a benchmarking experiment; see previous literature [21] for theoretical and empirical investigation of this problem. Therefore, "fishing for datasets" after completion of the benchmark experiment should be prohibited, see Rule 4 of the "ten simple rules for reducing over-optimistic reporting" [22].

Independent of the problem of fishing for significance, it is important that the criteria for inclusion in the benchmarking experiment are clearly stated as recently discussed [10]. In our study, we consider simple datasets' characteristics presented in Table 3.2. Based on these datasets' characteristics, we define several sets of inclusion criteria and investigate the impact of these choices 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.

| **Meta-Feature** | **Description** |
|---|---|
| $n$ | number of observations |
| $p$ | number of features |
| $\frac{p}{n}$ | dimensionality |
| $d$ | number of features of the associated design matrix for LR |
| $\frac{d}{n}$ | dimensionality of the design matrix |
| $p_{numeric}$ | number of numeric features |
| $p_{categorical}$ | number of categorical features |
| $p_{numeric,rate}$ | proportion of numeric features |
| $C_{max}$ | percentage of observation of the majority class |
| $time$ | duration for the run a 5-fold CV with a default Random Forest |

Table 1: Considered meta-features.

## 3.3 Meta-learning

Taking another perspective on the problem of benchmarking results being dependent on dataset's characteristics, we also consider modelling the difference between the methods' performances (considered as response variable) based on the datasets' characteristics (considered as features). Such a modelling approach can be seen as a simple form of *meta-learning*—a well-known task in machine

learning [23]. A similar approach using linear mixed models has been recently applied to the selection of an appropriate classification method in the context of high-dimensional gene expression data analysis [24]. Considering the potentially complex dependency patterns between response and features, we use RF as a prediction tool for this purpose.

### 3.4 Power calculation

Considering the $M \times 2$ matrix, collecting the performance measures for the two investigated methods (LR and RF) on the $M$ considered datasets, one can perform a test for paired samples to compare the performances of the two methods [25]. We refer to the previously published statistical framework [25] for a precise mathematical definition of the tested null-hypothesis in the case of the t-test for paired samples. In this framework, the datasets play the role of the *i.i.d.* observations used for the t-test. Sample size calculations for the t-test for paired samples can give an indication of the rough number of datasets required to detect a given difference $\delta$ in performances considered as relevant for a given significance level (e.g., $\alpha = 0.05$) and a given power (e.g., $1 - \beta = 0.8$). For large numbers and a two-sided test, the required number of datasets can be approximated as

$$M_{req} \approx \frac{(z_{1-\alpha/2} + z_{1-\beta})^2 \sigma^2}{\delta^2} \tag{4}$$

where $z_q$ is the $q$-quantile of the normal distribution and $\sigma^2$ is the variance of the difference between the two methods' performances over the datasets, which may be roughly estimated through a pilot study or previous literature.

For example, the required number of datasets to detect a difference in performances of $\delta = 0.05$ with $\alpha = 0.05$ and $1-\beta = 0.8$ is $M_{req} = 32$ if we assume a variance of $\sigma^2 = 0.01$ and $M_{req} = 8$ for $\sigma^2 = 0.0025$. It increases to $M_{req} = 197$ and $M_{req} = 50$, respectively, for differences of $\delta = 0.02$.

### 3.5 Availability of Data and Materials

Several R packages are used to implement the benchmarking study: `mlr` (version 2.10) for higher abstraction and a simpler way to conduct benchmark studies [26], `OpenML` (version 1.2) for loading the datasets [27], and `batchtools` (version 0.9.2) for parallel computing [28]. Note that the LR and RF learners called via `mlr` are wrappers on the functions `glm` and `randomForest`, respectively.

The datasets supporting the conclusions of this article are freely available in OpenML as described in 3.1.

Emphasis is placed on the reproducibility of our results. Firstly, the code implementing all our analyses is fully available from GitHub [29]. For visualization-only purposes, the benchmarking results are available from this link, so that our graphics can be quickly generated by mouse-click. However, the code to recompute these results, i.e. to conduct the benchmarking study, is also available from GitHub. Secondly, since we use a specific version of R and our results may

thus be difficult to reproduce in the future due to software updates, we also provide a docker image [30]. Docker automates the deployment of applications inside a so called "Docker container" [31]. We use it to create an R environment with all the packages we need in their correct version. Note that docker is not necessary here (since all our codes are available from GitHub), but very practical for a reproducible environment and thus for reproducible research in the long term.

# 4 Results

In our study we consider a set of $M$ datasets (see Section 4.1 for more details) and compute for each of them the performance of random forest and logistic regression according to the three performance metrics outlined in Section 2.4.

## 4.1 Included datasets

From approximately 20000 datasets currently available from OpenML [20], we select those featuring binary classification problems. Further, we remove the datasets that include missing values, the obviously simulated datasets as well as duplicated datasets. We also remove datasets with more features than observations ($p > n$), and datasets that require too much computation time, i.e. datasets such that $n \cdot p > 3 \cdot 10^6$ as they correspond exactly to the datasets such that one iteration of the 5-CV repeated 10 times takes more than 100s. This finally leaves us with a total of 278 datasets. Of these 278 datasets, 15 produced NAs for LR, and 3 did so for both LR and RF (more details on this problem are given in the next section), which finally leaves us with 260 datasets for our analysis.

## 4.2 Missing values due to errors

In Section 4.1 we stated that among the 278 datasets which we used for the benchmark, 18 produced an NA and were discarded for this reason (see also the flow-chart in Figure 2). We now give more details on these NAs in Table 4.2. Both LR and RF fail in the presence of categorical features with too many categories. More precisely, RF fails when more than 53 categories are detected in at least one of the features, while LR fails when levels undetected during the training phase occur in the test data. We could admittedly have prevented these errors through basic preprocessing of the data such as the removal of the features that induced errors. However, the decision was taken to just remove the datasets resulting in NAs because we did not want to address preprocessing steps, which would be a topic on their own and cannot be adequately treated along the way for such a high number of datasets. Note that in doing this kind of "complete case" analysis, we ignore an inconvenience of LR—that failed more often than its competitor RF. As an alternative strategy to handle NAs, we could also have decided to replace them by the worst possible performance

achievable on the considered dataset (in the case of more than 2 compared methods, it is also possible to replace NAs by the performance achieved on this dataset by the worst algorithm [32]). To conclude, in our benchmark study LR is inferior to RF on average (see Section 4.3) in terms of accuracy *even if* it is indirectly advantaged by our strategy to handle NAs.

| **Dataset's ID** | **RF fails** | **LR fails** |
|---|---|---|
| 3 | N | Y |
| 461 | N | Y |
| 463 | N | Y |
| 465 | Y | Y |
| 796 | N | Y |
| 825 | Y | Y |
| 865 | N | Y |
| 891 | N | Y |
| 938 | N | Y |
| 941 | N | Y |
| 942 | N | Y |
| 953 | N | Y |
| 1006 | N | Y |
| 1012 | N | Y |
| 1116 | Y | Y |
| 1167 | N | Y |
| 1470 | N | Y |
| 1506 | N | Y |

Table 2: Benchmark Errors for Random Forest and Logistic Regression.

## 4.3 Main results

Overall performances are presented in a synthesized form in Table 4.3 for all three measures in form of average performances along with standard deviations and confidence intervals computed using the adjusted bootstrap percentile (BCa) method [33]. The boxplots of performances of Random Forest (RF) and Logistic Regression (LR) for the three considered performance measures are depicted in Figure 3, which also includes the boxplot of the difference in performances (bottom row). It can be seen from Figure 3 that RF performs better in most of the cases (69.2 % of our datasets for *acc*, 73.8 % for *auc* and 71.2 % for *brier*). Furthermore, when LR outperforms RF the difference is minimal. It can also be noted that the differences in performance tend to be larger for *auc* than for *acc* and *brier*.

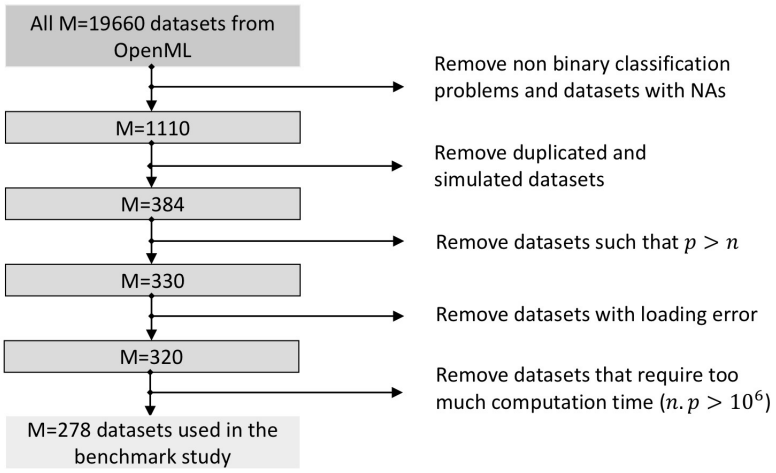


Figure 2: Selection criteria for datasets
Flowchart representing the criteria for selection of the datasets.

## 4.4 Explaining differences: datasets' characteristics

### 4.4.1 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.

As a preliminary, let us illustrate this idea using only one (large) dataset, the OpenML dataset with $ID = 1496$ including $n_0 = 7400$ observations and $p_0 = 20$ features. A total of $N = 50$ sub-datasets are extracted from this dataset by randomly picking a number $n' < n_0$ of observations or a number $p' < p_0$ of features. Thereby we choose $n'$ such that $\frac{n'}{n_0}$ successively takes the values $\frac{n'}{n_0}' = 0.20, 0.34, 0.48, 0.62, 0.76, 0.90$ and $p'$ such that $\frac{p'}{p_0}$ successively takes the values $\frac{p'}{p_0} = 0.20, 0.32, 0.44, 0.56, 0.68, 0.80$. Figure 4 displays the boxplots of the accuracy of RF and LR for varying $p'$ (top-left) and varying $n'$ (top-right). Each boxplot represents $N = 50$ data points.

It can be seen from Figure 4 that the accuracy increases with $p'$ for both LR and RF. This reflects the fact that relevant features may be missing from the considered random subsets $p'$ features. Thus, accuracy increases as more and

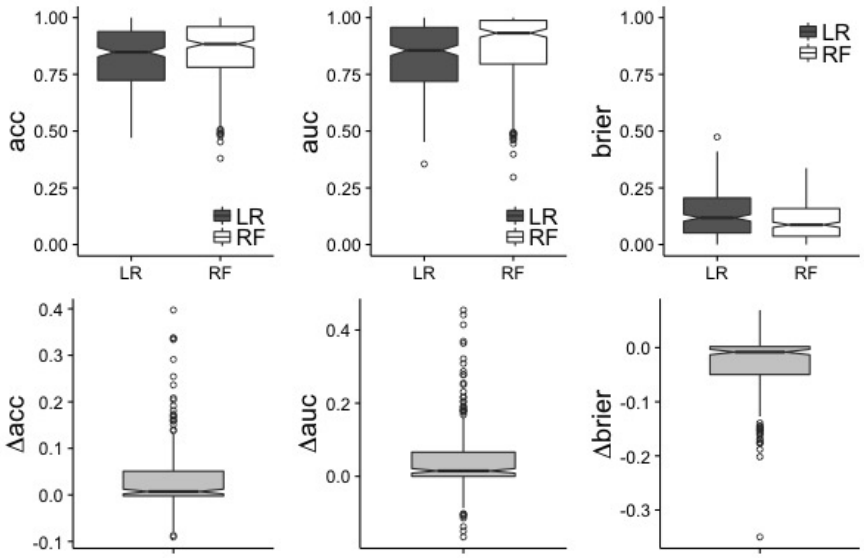


Figure 3: Main results of the benchmark experiment
Boxplots of the performance for the three considered measures on the 260 considered datasets. Top: boxplot of the performance of LR (dark) and RF (white) for each performance measure. Bottom: boxplot of the difference of performances $\Delta perf = perf_{RF} - perf_{LR}$.

more features are included. Interestingly, it can also be seen that the increase of accuracy with $p'$ is more pronounced for RF than for LR in the small $p'$ range, and vice-versa in the large $p'$ range. As a result, the difference in accuracy between RF and LR first increases with $p'$ and then (slightly) decreases, as can be seen from the bottom-left part of Figure 4. The increase in the small $p'$ range reflects the commonly formulated assumption that RF performs particularly well for data with a large number of features, while the decrease in the large $p'$ range is more difficult to explain. In contrast, as $n$ increases the difference in performances between RF and LR increases slightly but monotonously, while—as expected—its variance decreases; see the bottom-right part of Figure 4.

### 4.4.2 Subgroup analyses

To further explore this issue over all 260 investigated datasets, we computed Spearman's correlation coefficient between the difference in accuracy between random forest and logistic regression ($\Delta acc$) and various datasets' characteristics. The results of Spearman's correlation test are shown in Table 4.4.2. These analyses again point to the importance of the number $p$ of features (and related characteristics), while the dataset size $n$ and the percentage $C_{max}$ of observa-

| **Accuracy** | $\mu$ | $\sigma$ | BCa confidence interval |
|---|---|---|---|
| Logistic regression | 0.820 | 0.139 | [0.803, 0.837] |
| Random forest | 0.852 | 0.136 | [0.834, 0.869] |
| Difference | 0.032 | 0.071 | [0.024, 0.042] |
| **auc** | | | |
| Logistic regression | 0.824 | 0.152 | [0.806, 0.842] |
| Random forest | 0.867 | 0.151 | [0.847, 0.883] |
| Difference | 0.043 | 0.094 | [0.032, 0.055] |
| **Brier Score** | | | |
| Logistic regression | 0.131 | 0.093 | [0.120, 0.142] |
| Random forest | 0.103 | 0.081 | [0.093, 0.113] |
| Difference | -0.028 | 0.055 | [-0.036, -0.022] |

Table 3: Performances of LR and RF (top: accuracy, middle: AUC, bottom: Brier score): mean performance values $\mu$, standard deviation $\sigma$ and confidence intervals for the mean (estimated via the bootstrap BCa method [33]) on the 260 datasets.

tions in the majority class are not significantly correlated with $\Delta acc$.

To investigate these dependencies more deeply, we examine the performances of RF and LR within subgroups of datasets defined based on datasets' characteristics (called meta-features from now on), following the principle of subgroup analyses well-known in clinical research. As some of the meta-features displayed in Table 4.4.2 are mutually (highly) correlated, we cluster them using a hierarchical clustering algorithm (data not shown). From the resulting dendogram we decide to select the meta-features $p$, $n$, $\frac{p}{n}$, $C_{max}$, while other meta-features are considered redundant and ignored in further analyses.

| | **Spearman's $\rho$** | **Spearman's $\rho$ p-value** |
|---|---|---|
| $n$ | 0.0819 | $1.88 \cdot 10^{-1}$ |
| $p$ | 0.323 | $9.68 \cdot 10^{-8}$ |
| $\frac{p}{n}$ | 0.102 | $1.02 \cdot 10^{-1}$ |
| $d$ | 0.267 | $1.26 \cdot 10^{-5}$ |
| $\frac{d}{n}$ | 0.112 | $7.20 \cdot 10^{-2}$ |
| $p_{numeric}$ | 0.239 | $1.01 \cdot 10^{-4}$ |
| $p_{categorical}$ | -0.071 | $2.56 \cdot 10^{-1}$ |
| $p_{numeric,rate}$ | 0.227 | $2.20 \cdot 10^{-4}$ |
| $C_{max}$ | 0.017 | $7.79 \cdot 10^{-1}$ |

Table 4: Correlation between $\Delta acc$ and dataset's features.

Figure 5 displays the boxplots of the differences in accuracy for different subgroups based on the four selected meta-features $p$, $n$, $\frac{p}{n}$ and $C_{max}$. For each of the four meta-features, subgroups are defined based on different cut-off values,

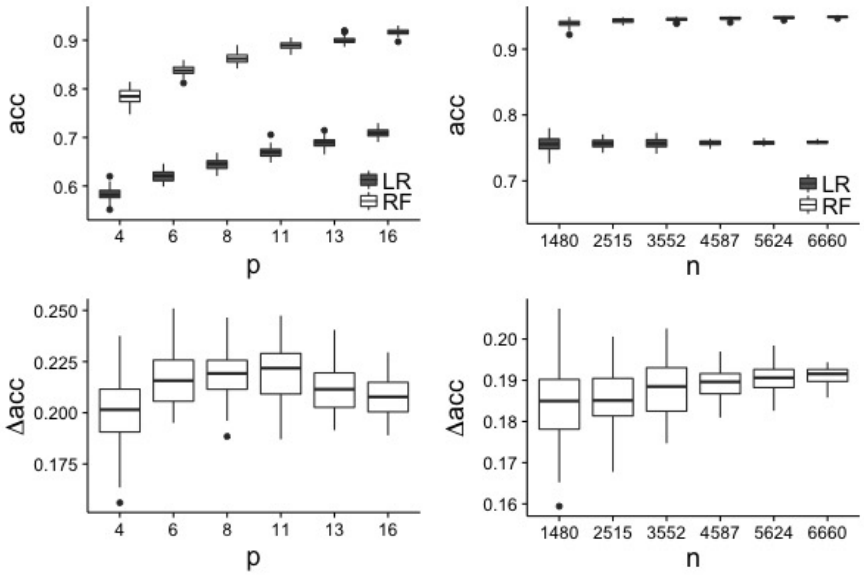


Figure 4: Influence of $n$ and $p$: subsampling experiment based on dataset ID=1496
Top: Boxplot of the performance ($acc$) of RF and LR for $N = 50$ sub-datasets extracted from the OpenML dataset with ID=1496 by randomly picking $n' \leq n$ observations and $p' < p$ features. Bottom: Boxplot of the differences in performances $\Delta acc = Acc_{RF} - Acc_{LR}$ between RF and LR. $p' \in \{4, 6, 8, 11, 13, 16\}$. $n' \in \{1480, 2515, 3552, 4587, 5624, 6660\}$. Performance is evaluated through 5-fold-cross-validation repeated 2 times.

denoted as $t$, successively. The histograms of the four meta-features for the 240 datasets are depicted in the bottom row of the figure, where the considered cutoff values are materialized as vertical lines. Similar pictures are obtained for the two alternative performance measures $auc$ and $brier$; See supplementary file 1.

It can be observed from Figure 5 that RF tends to yield better results than LR for a low $n$, and that the difference decreases with increasing $n$. In contrast, RF performs comparatively poorly for datasets with $p < 5$, but better than LR for datasets with $p > 5$. This is due to low performances of RF on a high proportion of the datasets with $p < 5$. For $\frac{p}{n}$, the difference between RF and LR is negligeable in low dimension ($\frac{p}{n} < 0.01$), but increases with the dimension. The contrast is particularly striking between the subgroups $\frac{p}{n} < 0.1$ (yielding a small $\Delta acc$) and $\frac{p}{n} > 0.1$ (yielding a high $\Delta acc$), again confirming the hypothesis that the superiority of RF over LR is more pronounced in high dimensional settings.

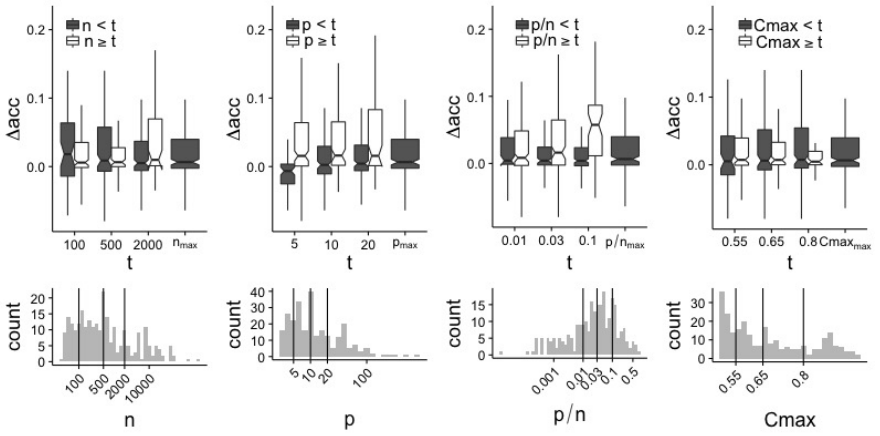


Figure 5: Subgroup analyses
Top: for each one of the four selected meta-features, boxplots of $\Delta acc$ for different threshold as criteria for dataset's selection. On the bottom we represent the distribution of our meta-features (log scale) to locate the chosen threshold. Note that outliers are not shown here for a more convenient visualization. For a corresponding figure including the outliers as well as the results for auc and brier, see supplementary file 1.

### 4.4.3 Meta-learning

The previous section showed that benchmarking results in subgroups may be considerably different from that of the entire datasets collection. Going one step further, one can extend the analysis of meta-features towards meta-learning to gain insight on their influence. More precisely, taking the datasets as observations we build a regression RF that predicts the difference in performance between RF and LR based on the four meta-features considered in the previous subsection ($p$, $n$, $\frac{p}{n}$ and $C_{max}$). Figure 6 depicts partial dependence plots for visualization of the influence of each meta-feature. Again, we notice a dependency on $p$ and $\frac{p}{n}$ as outlined in Section 4.4.2 and the comparatively bad results of RF when compared to LR for datasets with small $p$. The importance of $C_{max}$ and $n$ is less noticeable.

Although these results should be considered with caution, since they are possibly highly dependent on the particular distribution of the meta-features over the 240 datasets, we conclude from Section 4.4 that meta-features substantially affect $\Delta acc$. This points out the importance of a clear inclusion criteria definition for datasets in a benchmark experiment and of the consideration of the meta-features' distributions.

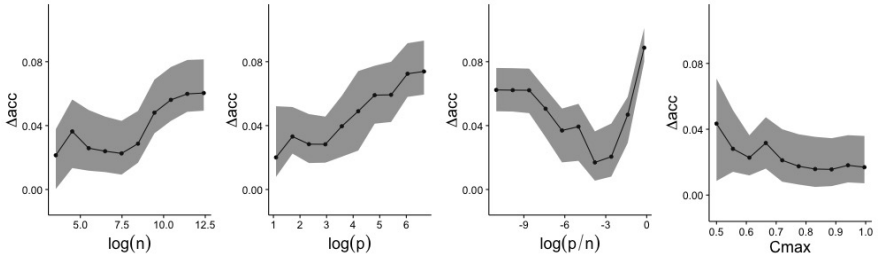


Figure 6: Meta-learning results
Plot of the partial dependence for the 4 considered meta-features : $log_(n)$, $log_(p)$, $log_(\frac{p}{n})$, $C_{max}$. The *log* scale was chosen for 3 of the 4 features to obtain more uniform distribution (see Figure 5 where the distribution is plotted in *log* scale). For each plot, the black line denotes the median of the individual partial dependances, and the lower and upper curves of the grey regions represent respectively the 25%- und 75%-quantiles. Estimated mse is 0.00426 via a 5-CV repeated 4 times.

### 4.5 Explaining differences: partial dependence plots

In the previous section we investigated the impact of datasets' characteristics on the results of benchmarking and modeled the difference between methods' performance based on these characteristics, termed "meta-features". In this section, we take a different approach for the explanation of differences. We use partial dependence plots as a technique to assess the dependency pattern between response and features underlying the prediction rule. More precisely, the aim of these additional analyses is to assess whether differences in performances (between LR and RF) are related to differences in partial dependence plots. After getting a global picture for all datasets included in our study, we inspect three interesting "extreme cases" more closely. In a nutshell, we observe no strong correlation between the difference in performances and the difference in partial dependences over the 260 considered datasets. More details are given in supplementary file 2.

## 5 Discussion

### 5.1 Summary

We presented a large-scale benchmark experiment for comparing the performance of logistic regression and random forest in binary classification settings. The overall results on our 240 datasets collection showed better for random forest (70% of the cases) than logistic regression (30%). On the whole, our results support the increasing use of RF with default parameter values as a standard method—which of course neither means that it performs better on all datasets nor that other parameter values/variants than the default are useless!

We devoted particular attention to the inclusion criteria applied when selecting datasets for our study. We investigated how the conclusions of our benchmark experiment change when varying the applied inclusion criteria. Our analyses reveal a noticeable influence of the number of features $p$ and the ratio $\frac{p}{n}$. The superiority of RF tends to be more pronounced for increasing $p$ and $\frac{p}{n}$. More generally, our study outlines the importance of inclusion criteria and the necessity to include a large number of datasets in benchmark studies as outlined in previous literature [10, 22, 25].

## 5.2 Limitations

Firstly, as previously discussed [10], results of benchmarking experiments should be considered as conditional on the set of included datasets. As demonstrated by our analyses on the influence of inclusion criteria for datasets, different sets of datasets yield different results. While the set of datasets considered in our study has the major advantages of being large and including datasets from various scientific fields, it is not strictly speaking representative of a "population of datasets", hence essentially yielding conditional conclusions.

Secondly, other aspects of classification methods are important but have not been considered in our study, for example issues related to the *transportability* of the constructed prediction rules. By transportability, we mean the possibility for interested researchers to apply a prediction rule presented in the literature to their own data [8, 9]. With respect to transportability, LR is clearly superior to RF, since it is sufficient to know the fitted values of the regression coefficient in application to a LR-based prediction rule. LR also has the major advantage that it yields interpretable prediction rules: it does not only aim at *predicting* but also at *explaining* effect, an important distinction that is extensively discussed elsewhere [1] and related to the "two cultures" of statistical modelling described by Leo Breiman [34]. These important aspects are not taken into account in our study, which deliberately focuses on prediction accuracy.

Thirdly, our study was intentionally restricted to RF with default values. The superiority of RF may be more pronounced if used together with an appropriate tuning strategy. Moreover, the version of RF considered in our study has been shown to be (sometimes strongly) biased in variable selection [13]. More precisely, variables of certain types (e.g., categorical variables with a large number of categories) are systematically preferred by the algorithm for inclusion in the trees irrespectively of their relevance for prediction. Variants of RF addressing this issue [12] may perform better, at least in some cases.

## 5.3 Outlook

In this paper, we fully ignore the problem of parameter tuning by simply setting the parameters to their default values as implemented in the widely used package `randomForest`. The rationale for this choice was to provide evidence for default values and thereby the analysis strategy most researchers currently apply in practice. The development of reliable and practical parameter tuning

strategies, however, is crucial and more attention should be devoted in the future. Parameter tuning may substantially improve the performance of RF in some cases [16, 35]. Particular attention should be given to the development of user-friendly tools, considering that one of the main reasons for using default values is probably the ease-of-use—an important aspect in the hectic academic context.

Any tuning strategy claimed to be a good candidate in becoming a "standard strategy" should ideally be subjected to a large-scale benchmarking experiment inspired from the study presented in this paper. By presenting the results on the average superiority with default values over LR, we by no means want to definitively establish these default values. Instead, our study is intended as a fundamental first step towards well-designed studies providing solid well-delimited evidence on the performance.

Before further studies are performed on tuning strategies, we insist that, whenever performed in applications of RF, parameter tuning should ideally always be reported clearly including all technical details either in the main or in its supplementary materials. Furthermore, the uncertainty regarding the "best tuning strategy" should in no circumstances be exploited for conscious or subconscious "fishing for significance".

# 6 Conclusion

Our systematic large-scale comparison study based on 260 real datasets shows the average good prediction performance of random forest (compared to logistic regression) even with the standard implementation and default parameters, which is in some respects suboptimal. This study should in our view be seen both as (i) an illustration of the application of principles borrowed from clinical trial methodology to benchmarking in computational sciences—an approach that could be more widely adopted in this field and (ii) a motivation to pursue research on random forests not only including possibly better variants and parameter choices but also strategies to improve their transportability.

# List of abbreviations

Acc: accuracy
auc: area under the curve
Brier: Brier score
CV: cross-validation
LR: logistic Regression
PDP: partial dependence plot
RF: random forest
VIM: variable importance measure

## Funding

This project was supported by the Deutsche Forschungsgemeinschaft (DFG), grants BO3139/6-1 and BO3139/2-3 to ALB.

## Acknowledgements

The authors thank Bernd Bischl for valuable comments and Jenny Lee for language corrections.

## References

[1] Shmueli, G.: To explain or to predict? Statistical Science **25**, 289–310 (2010)

[2] Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)

[3] Liaw, A., Wiener, M.: Classification and regression by randomforest. R News **2**, 18–22 (2002)

[4] Boulesteix, A.-L., Lauer, S., Eugster, M.J.: A plea for neutral comparison studies in computational sciences. PLOS ONE **8**(4), 61562 (2013)

[5] De Bin, R., Janitza, S., Sauerbrei, W., Boulesteix, A.-L.: Subsampling versus bootstrapping in resampling-based model selection for multivariable regression. Biometrics **72**, 272–280 (2016)

[6] Boulesteix, A.-L., De Bin, R., Jiang, X., Fuchs, M.: IPF-LASSO: integrative L1-penalized regression with penalty factors for prediction based on multi-omics data. Computational and Mathematical Models in Medicine (accepted) (2017)

[7] Boulesteix, A.-L., Bender, A., Bermejo, J.L., Strobl, C.: Random forest gini importance favours snps with large minor allele frequency: impact, sources and recommendations. Briefings in Bioinformatics **13**(3), 292–304 (2012)

[8] Boulesteix, A.-L., Schmid, M.: Machine learning versus statistical modeling. Biometrical Journal **56**(4), 588–593 (2014)

[9] Boulesteix, A.-L., Janitza, S., Hornung, R., Probst, P., Busen, H., Hapfelmeier, A.: Making complex prediction rules applicable for readers: Current practice in random forest literature and recommendations. Technical Report 199, Department of Statistics, LMU (2016)

[10] Boulesteix, A.-L., Wilson, R., Hapfelmeier, A.: Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies. Technical Report 198, Department of Statistics, LMU (2016)

[11] Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of Statistics **29**, 1189–1232 (2001)

[12] Hothorn, T., Hornik, K., Zeileis, A.: Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics **15**, 651–674 (2006)

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

[14] Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning **63**(1), 3–42 (2006)

[15] Boulesteix, A.-L., Janitza, S., Kruppa, J., König, I.R.: Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **2**(6), 493–507 (2012)

[16] Huang, B.F., Boutros, P.C.: The parameter sensitivity of random forests. BMC Bioinformatics **17**, 331 (2016)

[17] Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling methods for meta-model validation with recommendations for evolutionary computation. Evolutionary Computation **20**(2), 249–275 (2012)

[18] Lichman, M.: UCI Machine Learning Repository (2013). `http://archive.ics.uci.edu/ml`

[19] Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G.G., *et al.*: Arrayexpress—a public repository for microarray gene expression data at the ebi. Nucleic Acids Research **31**, 68–71 (2003)

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

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

[22] Boulesteix, A.-L.: Ten simple rules for reducing overoptimistic reporting in methodological computational research. PLoS Comput Biol **11**(4), 1004191 (2015)

[23] Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Machine learning **54**(3), 187–193 (2004)

[24] Jong, V.L., Novianti, P.W., Roes, K.C., Eijkemans, M.J.: Selecting a classification function for class prediction with gene expression data. Bioinformatics **32**, 1814–1822 (2016)

[25] Boulesteix, A.-L., Hable, R., Lauer, S., Eugster, M.J.: A statistical framework for hypothesis testing in real data comparison studies. The American Statistician **69**(3), 201–212 (2015)

[26] Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Jones, Z., Casalicchio, G.: Mlr: Machine Learning in R. (2016). R package version 2.10. `https://github.com/mlr-org/mlr`

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

[28] Lang, M., Bischl, B., Surmann, D.: batchtools: Tools for R to work on batch systems. The Journal of Open Source Software **2**(10) (2017). doi:10.21105/joss.00135

[29] Couronne, R., Probst, P.: (2017). doi:10.5281/zenodo.439090. `https://doi.org/10.5281/zenodo.439090`

[30] Couronne, R., Probst, P.: Docker image: Benchmarking random forest: a large- scale experiment (2017). doi:10.5281/zenodo.804427. `https://doi.org/10.5281/zenodo.804427`

[31] Boettiger, C.: An introduction to docker for reproducible research. SIGOPS Oper. Syst. Rev. **49**(1), 71–79 (2015). doi:10.1145/2723872.2723882

[32] Bischl, B., Schiffner, J., Weihs, C.: Benchmarking local classification methods. Computational Statistics **28**(6), 2599–2619 (2013)

[33] Davison, A.C., Hinkley, D.V.: Bootstrap Methods and Their Application. Cambridge University Press, Cambridge (1997)

[34] Breiman, L.: Statistical modeling: The two cultures (with comments and a rejoinder by the author). Statistical Science **16**(3), 199–231 (2001)

[35] Goldstein, B.A., Polley, E.C., Briggs, F.: Random forests for genetic association studies. Statistical Applications in Genetics and Molecular Biology **10**(1), 32 (2011)