Statistically Valid Variable Importance Assessment through Conditional Permutations (Supplementary Material)

Anonymous Author(s)
Affiliation
Address
email

A Conditional Permutation Importance (CPI) Wald statistic asymptotically controls type-I errors: hypotheses, theorem and proof

Outline The proof relies on the observation that the importance score defined in (4) is 0 in the asymptotic regime, where the permutation procedure becomes a sampling step, under the assumption that variable j is not conditionally associated with y. Then all the proof focuses on the convergence of the finite-sample estimator to the population one. To study this, we use the framework developed in [Williamson et al., 2021]. Note that the major difference with respect to other contributions [Watson and Wright, 2021] is that the ensuing inference is no longer conditioned on the estimated learner $\hat{\mu}$. Next, we first restate the precise technical conditions under which the different importance scores considered are asymptotically valid, i.e. lead to a Wald-type statistic that behaves as a standard normal under the null hypothesis.

Notations Let $\mathcal{F}$ represent the class of functions from which a learner $\mu : \mathbf{x} \mapsto y$ is sought.

Let P_0 be the data-generating distribution and P_n is the empirical data distribution observed after drawing n samples (noted n_{train} in the main text; in this section, we denote it n to simplify notations). The separation between train and test samples is actually only relevant to alleviate some technical conditions on the class of learners used. $\mathcal{M}$ is the general class of distributions from which $P_1, \ldots, P_n, P_0$ are drawn. $\mathcal{R} := \{c(P_1 - P_2) : c \in [0, \infty), P_1, P_2 \in \mathcal{M}\}$ is the space of finite signed measures generated by $\mathcal{M}$. Let l be the loss function used to obtain μ. Given $f \in \mathcal{F}$, $l(f; P_0) = \int l(f(\mathbf{x}), y) P_0(\mathbf{z}) d\mathbf{z}$, where $\mathbf{z} = (\mathbf{x}, y)$. Let μ_0 denote a population solution to the estimation problem $\mu_0 \in \text{argmin}_{f \in \mathcal{F}} l(f; P_0)$ and $\hat{\mu}_n$ a finite sample estimate $\hat{\mu}_n \in \text{argmin}_{f \in \mathcal{F}} l(f; P_n) = \frac{1}{n} \sum_{(\mathbf{x},y) \in P_n} l(f(\mathbf{x}), y)$.

Let us denote by $\dot{l}(\mu, P_0; h)$ the Gâteaux derivative of $P \mapsto l(\mu, P)$ at P_0 in the direction $h \in \mathcal{R}$, and define the random function $g_n : \mathbf{z} \mapsto \dot{l}(\hat{\mu}_n, P_0; \delta_{\mathbf{z}} - P_0) - \dot{l}(\mu_0, P_0; \delta_{\mathbf{z}} - P_0)$, where $\delta_{\mathbf{z}}$ is the degenerate distribution on $\mathbf{z} = (\mathbf{x}, y)$.

Hypotheses

(A1) (Optimality) there exists some constant $C > 0$, such that for each sequence $\mu_1, \mu_2, \cdots \in \mathcal{F}$ given that $\|\mu_n - \mu_0\| \to 0, |l(\mu_n, P_0) - l(\mu_0, P_0)| < C\|\mu_n - \mu_0\|_{\mathcal{F}}^2$ for each n large enough.

(A2) (Differentiability) there exists some constant $\kappa > 0$ such that for each sequence $\epsilon_1, \epsilon_2, \cdots \in \mathbb{R}$ and $h_1, h_2, \cdots \in \mathcal{R}$ satisfying $\epsilon_n \to 0$ and $\|h_n - h_\infty\| \to 0$, it holds that

$$\sup_{\mu \in \mathcal{F} : \|\mu - \mu_0\|_{\mathcal{F}} < \kappa} \left| \frac{l(\mu, P_0 + \epsilon_n h_n) - l(\mu, P_0)}{\epsilon_n} - \dot{l}(\mu, P_0; h_n) \right| \to 0.$$

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

(A3) (Continuity of optimization) $\|\mu_{P_0+\epsilon h} - \mu_0\|_{\mathcal{F}} = O(\epsilon)$ for each $h \in \mathcal{R}$.

(A4) (Continuity of derivative) $\mu \mapsto \dot{l}(\mu, P_0; h)$ is continuous at μ_0 relative to $\|.\|_{\mathcal{F}}$ for each $h \in \mathcal{R}$.

(B1) (Minimum rate of convergence) $\|\hat{\mu}_n - \mu_0\|_{\mathcal{F}} = o_P(n^{-1/4})$.

(B2) (Weak consistency) $\int g_n(\mathbf{z})^2 dP_0(\mathbf{z}) = o_P(1)$.

(B3) (Limited complexity) there exists some P_0-Donsker class $\mathcal{G}_0$ such that $P_0(g_n \in \mathcal{G}_0) \to 1$.

Proposition (Theorem 1 in [Williamson et al., 2021]) If the above conditions hold, $l(\hat{\mu}_n, P_n)$ is an asymptotically linear estimator of $l(\mu_0, P_0)$ and $l(\hat{\mu}_n, P_n)$ is non-parametric efficient.

Let $P_0^\star$ be the distribution obtained by sampling the j-th coordinate of $\mathbf{x}$ from the conditional distribution of $q_0(x^j|\mathbf{x}^{-\mathbf{j}})$, obtained after marginalizing over y:

$$q_0(x^j|\mathbf{x}^{-\mathbf{j}}) = \frac{\int P_0(\mathbf{x}, y)dy}{\int P_0(\mathbf{x}, y)dx^j dy}$$

$P_0^\star(\mathbf{x}, y) = q_0(x^j|\mathbf{x}^{-\mathbf{j}}) \int P_0(\mathbf{x}, y)dx^j$. Similarly, let $P_n^\star$ denote its finite-sample counterpart. It turns out from the definition of $\hat{m}^j_{CPI}$ in Eq. 4 that $\hat{m}^j_{CPI} = l(\hat{\mu}_n, P_n^\star) - l(\hat{\mu}_n, P_n)$. It is thus the final-sample estimator of the population quantity $m^j_{CPI} = l(\hat{\mu}_0, P_0^\star) - l(\hat{\mu}_0, P_0)$.

Given that $\hat{m}^j_{CPI} = l(\hat{\mu}_n, P_n^\star) - l(\hat{\mu}_0, P_0^\star) - (l(\hat{\mu}_n, P_n) - l(\hat{\mu}_0, P_0)) + l(\hat{\mu}_0, P_0^\star) - l(\hat{\mu}_0, P_0)$, the estimator $\hat{m}^j_{CPI}$ is asymptotically linear and non-parametric efficient.

The crucial observation is that under the j-null hypothesis, y is independent of x^j given $\mathbf{x}^{-\mathbf{j}}$. Indeed, in that case $P_0(\mathbf{x}, y) = q_0(x^j|\mathbf{x}^{-\mathbf{j}})P_0(y|\mathbf{x}^{-\mathbf{j}})P_0(\mathbf{x}^{-\mathbf{j}})$ and $P_0(x^j|\mathbf{x}^{-\mathbf{j}}, y) = P_0(x^j|\mathbf{x}^{-\mathbf{j}})$, so that $P_0^\star = P_0$. Hence, mean/variance of $\hat{m}^j_{CPI}$'s distribution provide valid confidence intervals for m^j_{CPI} and $mean(\hat{m}^j_{CPI}) \underset{n\to\infty}{\to} 0$. Thus, the Wald statistic $\hat{z}^j_{CPJ}$ defined in section (4.2) converges to a standard normal distribution, implying that the ensuing test is valid.

In practice, hypothesis (B3), which is likely violated, is avoided by the use of cross-fitting as discussed in [Williamson et al., 2021]: as stated in the main text, variable importance is evaluated on a set of samples not used for training. An interesting impact of the cross-fitting approach is that it reduces the hypotheses to (A1) and (A2), plus the following two:

(B'1) (Minimum rate of convergence) $\|\hat{\mu}_n - \mu_0\|_{\mathcal{F}} = o_P(n^{-1/4})$ on each fold of the sample splitting scheme.

(B2') (Weak consistency) $\int g_n(\mathbf{z})^2 dP_0(\mathbf{z}) = o_P(1)$ on each fold of the sample splitting scheme.

B Evaluation Metrics

AUC score [Bradley, 1997]: The variables are ordered by increasing p-values, yielding a family of p splits into relevant and non-relevant at various thresholds. AUC score measures the consistency of this ranking with the ground truth ($n_{signals}$ predictive features versus $p - n_{signals}$).

Type-I error : Some methods output p-values for each of the variables, that measure the evidence against each variable being a null variable. This score checks whether the rate of low p-values of null variables is not exceeding the nominal false positive rate (set to 0.05).

Power : This score reports the average proportion of informative variables detected (when considering variables with p-value < 0.05).

Computation time : The average computation time per core on 100 cores.

Prediction Scores : As some methods share the same core to perform inference and with the data divided into a train/test scheme, we evaluate the predictive power for the different cores on the test set.

C Supplement Figure 1 - Diagram of CPI

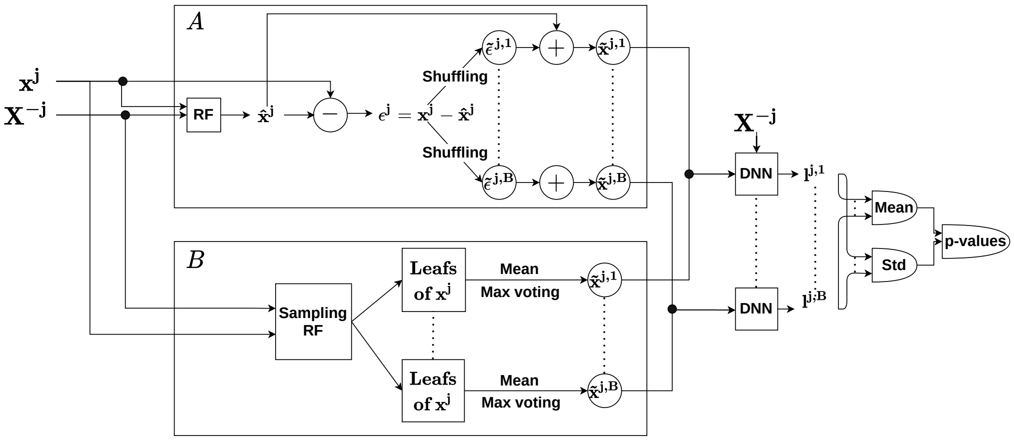
A
$\mathbf{x^j}$
$\mathbf{X^{-j}}$
RF
$\hat{\mathbf{x}}^{\mathbf{j}}$
$-$
$\epsilon^{\mathbf{j}} = \mathbf{x^j} - \hat{\mathbf{x}}^{\mathbf{j}}$
Shuffling
Shuffling
$\tilde{\epsilon}^{\mathbf{j},1}$
$\tilde{\epsilon}^{\mathbf{j},B}$
$+$
$+$
$\tilde{\mathbf{x}}^{\mathbf{j},1}$
$\tilde{\mathbf{x}}^{\mathbf{j},B}$
B
Sampling RF
Leafs of $\mathbf{x^j}$
Leafs of $\mathbf{x^j}$
Mean
Max voting
Mean
Max voting
$\tilde{\mathbf{x}}^{\mathbf{j},1}$
$\tilde{\mathbf{x}}^{\mathbf{j},B}$
$\mathbf{X^{-j}}$
DNN
DNN
$\mathbf{l^{j,1}}$
$\mathbf{l^{j,B}}$
Mean
Std
p-values

Figure 1: **CPI-DNN's constructions**: Constructing the variable of interest $\tilde{\mathbf{x}}^{\mathbf{j}}$ is done either (1) by the additive construction (top block) where a shuffled version of the residuals is added to the predicted version using the remaining predictors with the mean of a random forest (RF) or (2) by the sampling construction (bottom block) using a random forest (RF) model to fit $\mathbf{x^j}$ from $\mathbf{X^{-j}}$ and then sample the prediction within the leaves of the RF.

D Supplement Figure 1 - Power & Computation time

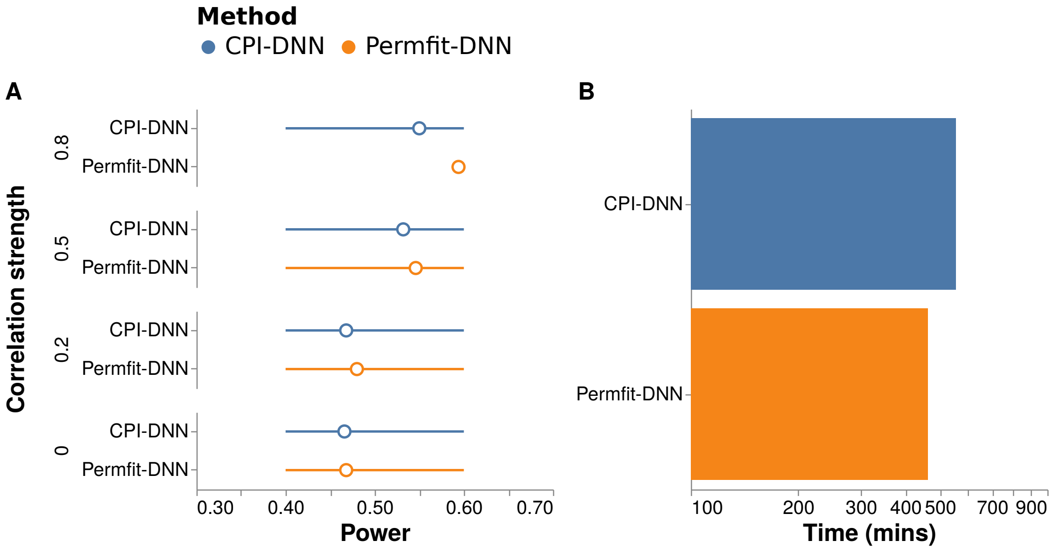

Figure 2: **Permfit-DNN vs CPI-DNN**: Performance at detecting important variables on simulated data under the setting of experiment 1, with n = 300 and p = 100. **(A)**: The power reports the average proportion of informative variables detected (p-value < 0.05). **(B)**: The computation time is in mins with (log10 scale) per core on 100 cores.

Based on Fig. 2, both methods *Permfit-DNN* and *CPI-DNN* have almost similar power. In high correlation regime, Permfit-DNN yields more detections, but it does not control type-I errors (Fig. 1). Regarding computation time, *CPI-DNN* is slightly more computationally expensive than *Permfit-DNN*.

E Supplement Figure 3 - Extended model comparisons

We also benchmarked the following methods deprived of statistical guarantees:

- Knockoffs [Candes et al., 2017, Nguyen et al., 2020]: The knockoff filter is a variable selection method for multivariate models that controls the False Discovery Rate. The first step of this procedure involves sampling extra null variables that have a correlation structure similar to that of the original variables. A statistic is then calculated to measure the strength of the original variables versus their knockoff counterpart. We call this the knockoff statistic $\mathbf{w} = \{w_j\}_{j=1}^{p}$ that is the difference between the importance of a given feature and the importance of its knockoff.
- Approximate Shapley values [Burzykowski, 2020]: SHAP being an instance method, we relied on an aggregation (averaging) of the per-sample Shapley values.
- Shapley Additive Global importancE (SAGE) [Covert et al., 2020]: Whereas SHAP focuses on the *local interpretation* by aiming to explain a model's individual predictions, SAGE is an extension to SHAP assessing the role of each feature in a *global interpretability* manner. The SAGE values are derived by applying the Shapley value to a function that represents the predictive power contained in subsets of features.
- Mean Decrease of Impurity [Louppe et al., 2013]: The importance scores are related to the impact that each feature has on the impurity function in each of the nodes.
- BART [Chipman et al., 2010]: BART is an ensemble of additive regression trees. The trees are built iteratively using a back-fitting algorithm such as MCMC (Markov Chain Monte Carlo). By keeping track of covariate inclusion frequencies, BART can identify which components are more important for explaining $\mathbf{y}$.

Based on AUC, we observe SHAP, SAGE and Mean Decrease of Impurity (MDI) perform poorly. These approaches are vulnerable to correlation. Next, Knockoff-Deep and Knockoff-Lasso perform well when the model does not include interaction effects. BART and Knockoff-Bart show fair performance overall.

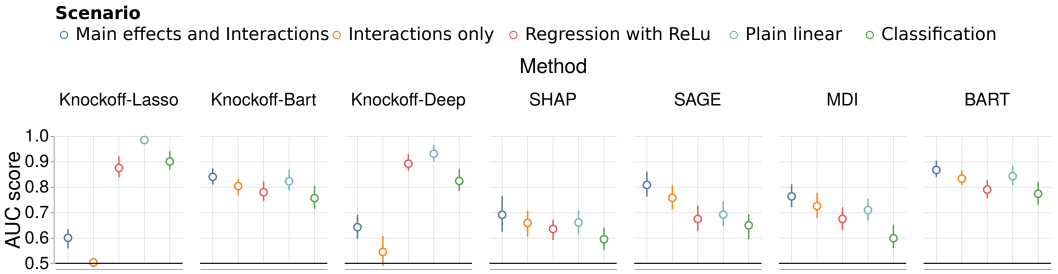

Figure 3: **Extended model comparisons**: State-of-the-art methods for variable importance not providing statistical guarantees in terms of p-values are compared (outer columns) and to competing approaches across data-generating scenarios (inner columns) using the settings of experiments 2 and 3. Prediction tasks were simulated with n = 1000 and p = 50. Solid line: chance level.

F Supplement Figure 3 - Power

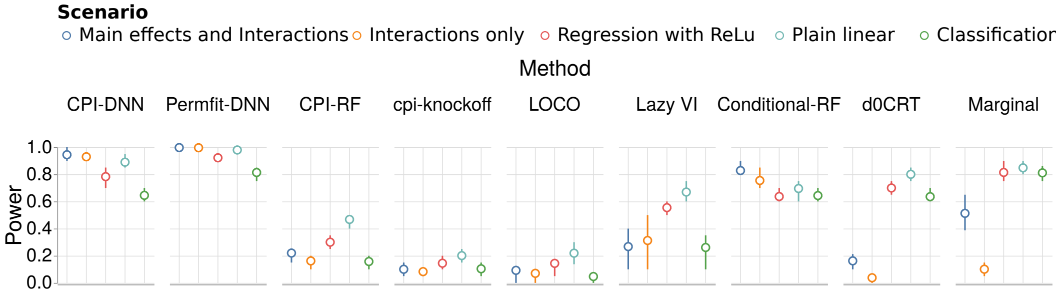

Figure 4: **Extended model comparisons**: *CPI-DNN* and *Permfit-DNN* were compared to baseline models (outer columns) and to competing approaches across data-generating scenarios (inner columns). Convention about power as in Fig. 2. Prediction tasks were simulated with n = 1000 and p = 50.

Based on the power computation, *Permfit-DNN* and *CPI-DNN* outperform the alternative methods. Thus, the use of the right learner leads to better interpretations.

G Supplement Figure 3 - Computation time

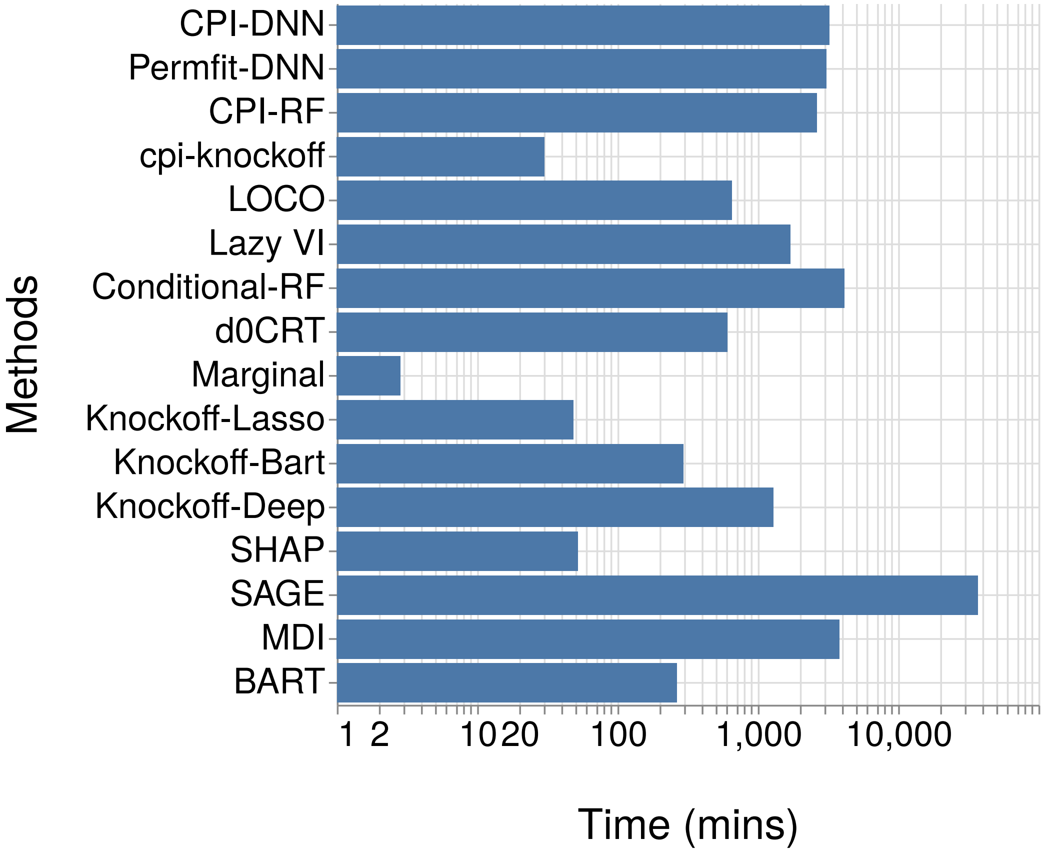

Figure 5: **Extended model comparisons**: The computation times for the different methods (with and without statistical guarantees in terms of p-values) are reported in mins with (log10 scale) per core on 100 cores. Prediction tasks were simulated with n = 1000 and p = 50.

The computation time of the different methods mentioned in this work (with and without statistical guarantees) is presented in Fig. 5 in mins with (log10 scale). First, we compare *CPI-RF*, *cpi-knockoff* and *LOCO* based on a Random Forest learner with p=50. We see that *cpi-knockoff* and *LOCO* are faster than *CPI-DNN*. A possible reason is that *CPI-DNN* uses an inner 2-fold internal validation for hyperparameter tuning (learning rate, L1 and L2 regularization) unlike the alternatives. Next, The DNN-based methods (*CPI-DNN* and *Permfit-DNN*) are competitive with the alternatives that control type-I error (d_0CRT, *cpi-knockoff* and *LOCO*) despite the use of computationally lean learners in the latter.

H Supplement Figure 3 - Prediction scores on simulated data

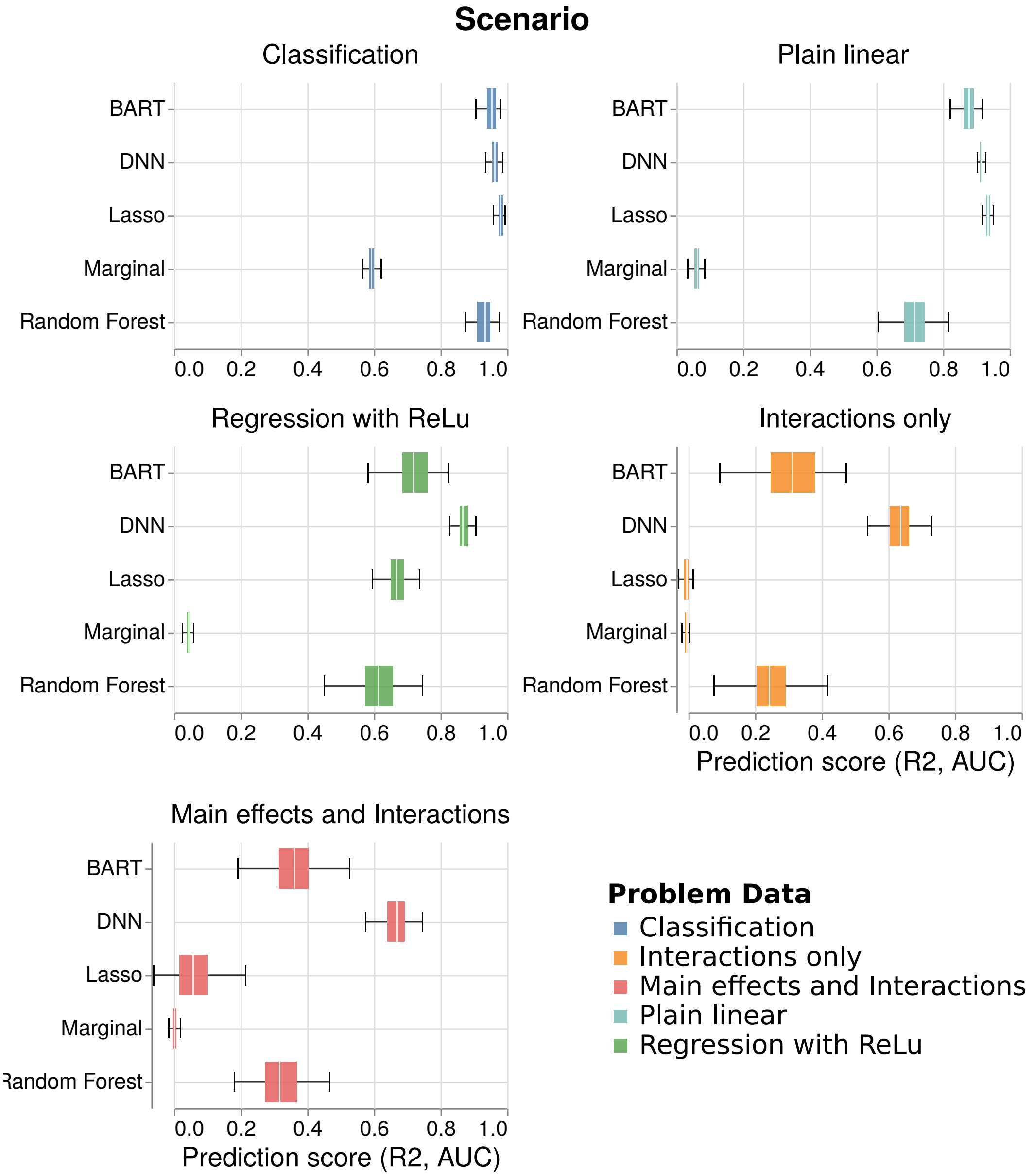

Figure 6: **Evaluating predictive power**: Performance of the different base learners used in the variable importance methods (**Marginal** = {Marginal effects}, **Lasso** = {Knockoff-Lasso}, **Random Forest** = {MDI, d0CRT, CPI-RF, Conditional-RF, cpi-knockoff, LOCO}, **BART** = {Knockoff-BART, BART} and **DNN** = {Knockoff-Deep, Permfit-DNN, CPI-DNN, Lazy VI}) on simulated data with n = 1000 and p = 50 in terms of **ROC-AUC** score for the classification and **R2** score for the regression.

The results for computing the prediction accuracy using the underlying learners of the different methods are reported in Fig. 6. Marginal inference, performs poorly, as it is not a predictive approach. Linear models based on Lasso show a good performance in the no-interaction effect scenario. Non-linear models based on Random Forest and BART improve on the lasso-based models. Nevertheless, they fail to achieve a good performance in scenarios with interaction effects. The models equipped with a deep learner outperform the other methods.

References

Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, July 1997. ISSN 0031-3203. doi: 10.1016/S0031-3203(96)00142-2.

Przemyslaw Biecek and Tomasz Burzykowski. *Explanatory Model Analysis*. December 2020.

Emmanuel Candes, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for Gold: Model-X Knockoffs for High-dimensional Controlled Variable Selection. *arXiv:1610.02351 [math, stat]*, December 2017.

Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1), March 2010. ISSN 1932-6157. doi: 10.1214/09-AOAS285.

Ian Covert, Scott Lundberg, and Su-In Lee. Understanding Global Feature Contributions With Additive Importance Measures, October 2020.

Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. page 9, January 2013.

Tuan-Binh Nguyen, Jérôme-Alexis Chevalier, Bertrand Thirion, and Sylvain Arlot. Aggregation of Multiple Knockoffs. *arXiv:2002.09269 [math, stat]*, June 2020.

David S. Watson and Marvin N. Wright. Testing Conditional Independence in Supervised Learning Algorithms, May 2021.

Brian D. Williamson, Peter B. Gilbert, Noah R. Simon, and Marco Carone. A general framework for inference on algorithm-agnostic variable importance, September 2021.