Interpretability and Explainability in Machine Learning (IML)

4. Model-agnostic interpretability methods

Pedro Delicado
Departament d'Estadística i Investigació Operativa and IMTech
Universitat Politècnica de Catalunya - Barcelona TECH

eBISS 2022. July 4-8, 2022, Cesena, Italy

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Introduction to model-agnostic interpretability methods

- Model-agnostic interpretation methods are those that only require the evaluation of the fitted prediction model on the training set, on the test set, or on perturbations of them.
- No other information from the kind of model at hand is needed.
- To be more formal, let f be the prediction function estimated from a training sample using a generic prediction model.
- We assume that f depends on p arguments, so the predicted value for $x = (x_1, \dots, x_p)$ is $f(x)$.
- The only connection between the model-agnostic interpretation methods and the prediction model is through the function f and, more specifically, only evaluations of f at different points x are allowed.

- Under this setting, to interpret the prediction model equals to interpret the prediction function f.
- This task is essentially the same we would have to do if we wanted to explore a generic mathematical function g depending on p variables, from which only evaluations are allowed.
- Therefore, any procedure that allows exploring a generic function g (using only evaluations of g) can be considered a model-agnostic method that could be used for interpreting a prediction function f.
- For instance, computing a numerical approximation to the gradient of f at a point x can be considered a model-agnostic interpretation method, as well as using this approximation to compute the first order Taylor expansion of f around x.

- Model-agnostic methods are specially useful for interpreting prediction models for which there are no specific interpretation methods.
- Nevertheless, model-agnostic methods can also improve the interpretation of models that are usually considered interpretable.
- Even multiple linear regression could benefit from the application of some model-agnostic methods.
- When a new model-agnostic interpretation method is introduced, a good practice is to check what it provides when applied to a classical simple prediction model, as linear regression, logistic regression or their additive extensions.
- In these simple cases, sometimes it is possible to obtain the closed expression of the new method results and then to relate them to the standard outputs of the classical methods.
- This way the new interpretation method will be either reinforced (when its classical counterpart is a sensible measure) or called into question (when the opposite happens).

We present several model-agnostic interpretation methods below, classified as global or local measures.

Classification of models and interpretability tools

Transparent models	Black-boxes: Post-modeling interpretability
Linear model (LM) GLM GAM CART Rule based models Naïve Bayes k-nearest neighbours	**Model-specific methods:** • Tree ensembles • Neural networks • Support vector machines **Model-agnostic methods:** Global measures • Variable importance by - Leave-one-covariate-out (LOCO) - Perturbing a variable in the test set: Random permutations, knockoffs, **Ghost-variables**, ... • Variable importance based on Shapley's value • Partial dependence plot (PDP) • Accumulated local effects plot (ALE) Local measures • Local interpretable model-agnostic explanations (LIME) • Local variable importance based on Shapley's value • SHAP (SHapley Additive exPlanations) • Break-down plots • Individual conditional expectation (ICE) plot, or ceteris paribus plot

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Global measures of variable relevance

- Let us consider the prediction problem involving the random vector (X, Z, Y), $X \in \mathbb{R}^p$, $Z \in \mathbb{R}$ and $Y \in \mathbb{R}$, where Y is considered the response variable that should be predicted from (X, Z).
- A prediction function $f : \mathbb{R}^{p+1} \longrightarrow \mathbb{R}$ has expected loss (or risk)

$$R(f(X, Z), Y) = \mathbb{E}(L(f(X, Z), Y)),$$

  where $L : \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}^+$ is a loss function measuring the cost associated with predicting Y by $f(X, Z)$.
- We consider the problem of measuring the effect of the single variable Z on the prediction function f when predicting Y by $f(X, Z)$:
  - **Variable relevance** or **variable importance** of Z.
- We assume that a training sample of size n_1 and a test sample of size n_2 are available.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Leave-one-covariate-out, LOCO

- A simple approach to define the importance of the variable Z:
  1. Fit the model including both X and Z.
  2. Fit the model including only X (leaving Z out).
  3. Relevance of Z by LOCO: The relative decrease in prediction accuracy in the test sample when Z is omitted from the model.
- This approach is used, for instance, in multiple linear regression to decide if the variable Z should be included in the model.
- The model must be fitted twice.

Relevance by LOCO, at a populational level

- It could happen that there would exist a natural reduced version of f, say f_p, depending only on p variables such that $f_p(X)$ would be the prediction of Y when Z is not available.
- For instance, the natural reduced version of $f(X,Z) = \beta_0 + X^T\beta_x + Z\beta_z$ could be $f_p(X) = \beta_0' + X^T\beta_x'$, for some β_0' and β_x', possibly different from β_0 and β_x, respectively.
- In this case, the usual relevance measure of Z is

  $$R(f_p(X), Y) - R(f(X,Z), Y)$$

  the reduction in the risk function when using Z.
- An alternative measure: $\mathbb{E}(L(f(X,Z), f_p(X)))$.

The case of quadratic loss

- Under quadratic loss, the first measure of relevance is

  $$R(f_p(X),Y)-R(f(X,Z),Y)=\mathbb{E}((Y-f_p(X))^2)-\mathbb{E}((Y-f(X,Z))^2),$$

  while the second equals

  $$\mathbb{E}(L(f(X,Z),f_p(X)))=\mathbb{E}((f(X,Z)-f_p(X))^2).$$

- Both measures coincide under quadratic loss, when $(Y-f(X,Z))$ has zero mean and it is independent of (X,Z):

  $$R(f_p(X),Y)=\mathbb{E}((Y-f_p(X))^2)=\mathbb{E}(\{(Y-f(X,Z))+(f(X,Z)-f_p(X))\}^2)=$$
  $$\mathbb{E}((Y-f(X,Z))^2)+\mathbb{E}((f(X,Z)-f_p(X))^2)+2\mathbb{E}(Y-f(X,Z))\mathbb{E}(f(X,Z)-f_p(X))=$$
  $$R(f(X,Z),Y)+\mathbb{E}(L(f(X,Z),f_p(X)))+0\Rightarrow$$
  $$R(f_p(X),Y)-R(f(X,Z),Y)=\mathbb{E}(L(f(X,Z),f_p(X))).$$

LOCO, additivity, linearity and quadratic loss

- Additive model: $Y = \beta_0 + s_1(X) + s_2(Z) + \varepsilon$

$$f(X, Z) = \mathbb{E}(Y|X, Z) = \beta_0 + s_1(X) + s_2(Z)$$

$$f_p(X) = \mathbb{E}(Y|X) = \beta_0 + s_1(X) + \mathbb{E}(s_2(Z)|X) = \beta_0 + s_1'(X)$$

  - Relevance of Z by LOCO:

$$\mathbb{E}(L(f(X, Z), f_p(X))) = \mathbb{E}((s_2(Z) - \mathbb{E}(s_2(Z)|X))^2) = \mathbb{E}(\mathrm{Var}(s_2(Z)|X))$$

- Under additional linearity: $Y = \beta_0 + X^T\beta_X + Z\beta_Z + \varepsilon$
  - Relevance of Z by LOCO:

$$\mathbb{E}(\mathrm{Var}(Z\beta_Z|X)) = \beta_Z^2\mathbb{E}(\mathrm{Var}(Z|X))$$

1 Introduction to model-agnostic interpretability methods

2 Global measures of variable relevance
- Leave-one-covariate-out, LOCO
- Variable importance by random permutations
- Relevance by ghost variables
- Other relevance measures based on perturbations
- The relevance matrix
- Variable relevance measures as Shapley's values
- Global graphical methods.

3 Local measures
- LIME
- Explaining individual predictions from Shapley values
- Local graphical methods

4 IML in R and Python

Variable importance by random permutations

In the context of random forests, Breiman (2001) proposed an alternative to LOCO: randomly permute the values of Z in the test sample (OOB):

1. Fit the model with the training sample using all the original explanatory variables, X and Z.
2. Evaluate the accuracy of the estimated model in the test sample using the observed values of X and Z.
3. Replace the values of Z in the test sample by a random permutation: Z'.
4. Evaluate the accuracy of the estimated model in the test sample using the observed values of X and the permuted values Z'.
5. Relevance of Z by random permutation: The relative decrease in prediction accuracy in the test sample when Z is replaced by Z'.

- Steps 3 and 4 can be repeated S times, and then the S accuracy measures are averaged in Step 5.
- The model is estimated only once.

Random permutations, at a population level

- The population counterpart of taking random permutations of values of Z in the test sample, is to replace the random variable Z by an independent copy of it, Z', with the same marginal distribution as Z but independent from (X, Y).
- This approach does not require the reduced version f_p of f.
- In this way the relevance measure for Z will be

$$\mathbb{E}(L(f(X,Z), f(X,Z'))).$$

Random permutations, additivity, linearity and quadratic loss

- Consider the case of f being additive in X and Z: $f(X,Z) = \beta_0 + s_1(X) + s_2(Z)$, with $\mathbb{E}(s_2(Z)) = 0$.
- Under quadratic loss, $\mathbb{E}(L(f(X,Z), f(X,Z')))=$

$$\mathbb{E}\left(\{(\beta_0 + s_1(X) + s_2(Z)) - (\beta_0 + s_1(X) + s_2(Z'))\}^2\right) = 2\text{Var}(s_2(Z)).$$

- If additional linearity happens, $s_2(Z) = Z\beta_z$, then this relevance measure of Z equals $2\beta_z^2\text{Var}(Z)$.

Random permutations: Undesirable properties.

At a first glance these relevance measures ($2\text{Var}(s_2(Z))$ or $2\beta_z^2\text{Var}(Z)$) seem to be suitable, but:

(1) The relevance of Z would be the same in two completely different cases:

- X and Z are independent; Z encode exclusive information about Y.
- X and Z are strongly related; in such a case X could make up for the absence of Z.

Clearly Z is more relevant in the first case than in the second one, but neither $2\text{Var}(s_2(Z))$ nor $2\beta_z^2\text{Var}(Z)$ can detect it.

(2) The replacement of Z by an independent copy Z' implies a drastic alteration of the prediction function $f(X, Z)$.

- Consider again the simple case of the linear predictor $f(X, Z) = \beta_0 + X^T\beta_x + Z\beta_z$.
- Replacing Z by Z' is equivalent to using the following reduced version of f:

  $$f_p(X) = \beta_0' + X^T\beta_x + \nu,$$

  where $\beta_0' = \beta_0 + \beta_z\mathbb{E}(Z)$ and $\nu = \beta_z(Z' - \mathbb{E}(Z))$, a zero mean random variable independent from (X, Y) that does not contribute in any way to the prediction of Y.
- A preferred alternative would be to use the reduced version of f given just by $\beta_0' + X^T\beta_x$, that is equivalent to replacing Z by $\mathbb{E}(Z)$ in $f(X, Z)$:

  $$f_p(X) = f(X, \mathbb{E}(Z)) = \beta_0 + X^T\beta_x + \mathbb{E}(Z)\beta_z = (\beta_0 + \mathbb{E}(Z)\beta_z) + X^T\beta_x.$$

(3) When X and Z are strongly related, there is a risk of extrapolation when evaluating $f(X, Z')$, because the support of (X, Z) could be much smaller than the support of (X, Z'), which is the Cartesian product of the supports of X and Z.

1 Introduction to model-agnostic interpretability methods

2 Global measures of variable relevance
  - Leave-one-covariate-out, LOCO
  - Variable importance by random permutations
  - Relevance by ghost variables
  - Other relevance measures based on perturbations
  - The relevance matrix
  - Variable relevance measures as Shapley's values
  - Global graphical methods.

3 Local measures
  - LIME
  - Explaining individual predictions from Shapley values
  - Local graphical methods

4 IML in R and Python

Relevance by ghost variables

- We have seen that replacing Z by $\mathbb{E}(Z)$ in $f(X,Z)$ is more appropriate than replacing it by an independent copy Z' (random permutations).
- But even better is to replace Z by $\mathbb{E}(Z|X)$: the best prediction of Z as a function of X, according to quadratic loss.
- If there is dependence between X and Z, we expect $|Z - \mathbb{E}(Z|X)|$ to be lower than $|Z - \mathbb{E}(Z)|$, so $f(X, \mathbb{E}(Z|X))$ is expected to be closer to $f(X,Z)$ than $f(X, \mathbb{E}(Z))$.
- Therefore, when Z is not available, replacing it by $\mathbb{E}(Z|X)$ allows X to contribute a little bit more in the prediction of Y than replacing Z by $\mathbb{E}(Z)$.
- The larger is this extra contribution of X, the smaller is the relevance of Z in the prediction of Y, measured by

$$\mathbb{E}(\, L(f(X,Z), f(X, \mathbb{E}(Z|X)))\,).$$

- We call **ghost variable** of Z to any estimator of $\mathbb{E}(Z|X)$.

Relevance by ghost variables (Delicado and Peña 2019)

1. Fit the model with the training sample using all the original explanatory variables, X and Z.
2. Evaluate the accuracy of the estimated model in the test sample using the observed values of X and Z.
3. Define the ghost variable for Z as $\hat{Z} = \widehat{\mathbb{E}(Z|X)}$, where the last estimation is done in the test sample.
4. Evaluate the accuracy of the estimated model in the test sample using the observed values of X and the ghost variable $\hat{Z}$.
5. Relevance of Z by its ghost variable: the relative decrease in prediction accuracy in the test sample when Z is replaced by $\hat{Z}$.

The ghost variables approach to measure the effect of variable Z combines the advantages of LOCO and random permutations:

- The model is estimated only once.
- It gives similar results to LOCO, which are better than those of random permutations when there are dependence among covariates.

Ghost variables, additivity, linearity and quadratic loss

- If f is additive in X and Z, under quadratic loss, the relevance of Z by its ghost variable is

$$\mathbb{E}(\, L(f(X,Z), f(X, \mathbb{E}(Z|X)))\,) = \mathbb{E}((s_2(Z) - s_2(\mathbb{E}(Z|X)))^2).$$

  It does not coincide neither with LOCO nor random permutations.

- If, additionally, there is linearity, $s_2(Z) = Z\beta_z$, it is equal to

$$\beta_z^2 \mathbb{E}((Z - \mathbb{E}(Z|X))^2) = \beta_z^2 \mathbb{E}(\mathrm{Var}(Z|X)),$$

  which coincides with the LOCO relevance of Z in this case.

- Relevance by random permutation gives a different measure: $\beta_z^2 \mathbb{E}(\mathrm{Var}(Z|X))$ coincides with $\beta_z^2 \mathrm{Var}(Z)$ when X and Z are independent, but otherwise the former would be preferred to the second as relevance measure of Z.

Variable relevance for a data set

- Consider the regression model $Y = m(X, Z) + \varepsilon$ and quadratic loss.
- A training sample of n_1 independent realizations of (X, Z, Y), $\mathcal{S}_1 = \{(x_{1.i}, z_{1.i}, y_{1.i}), i = 1, \ldots, n_1\}$, is used to estimate $m(x, z)$ as $\hat{m}(x, z)$ by a statistical or algorithmic procedure.
- A test sample $\mathcal{S}_2 = \{(x_{2.i}, z_{2.i}, y_{2.i}), i = 1, \ldots, n_2\}$ is available.
- Relevance by LOCO:

$$\mathrm{Rel}_{\mathrm{LOCO}}(Z) = \frac{1}{n_2} \sum_{i=1}^{n_2} (\hat{m}(x_{2.i}, z_{2.i}) - \hat{m}_p(x_{2.i}))^2.$$

- Relevance by a random permutation:

$$\mathrm{Rel}_{\mathrm{RP}}(Z) = \frac{1}{n_2} \sum_{i=1}^{n_2} (\hat{m}(x_{2.i}, z_{2.i}) - \hat{m}(x_{2.i}, z'_{2.i}))^2.$$

- Relevance by a ghost variable:

$$\mathrm{Rel}_{\mathrm{Gh}}(Z) = \frac{1}{n_2} \sum_{i=1}^{n_2} (\hat{m}(x_{2.i}, z_{2.i}) - \hat{m}(x_{2.i}, \hat{\mathbb{E}}(Z|X = x_{2.i})))^2.$$

Relevance measures in multiple linear regression

- If the relevance by LOCO is evaluated in the training sample, then

$$\mathrm{Rel}^{\mathrm{Train}}_{\mathrm{LOCO}}(Z) = \hat{\beta}_z^2 \hat{\sigma}^2_{[z]n_1} \Rightarrow \frac{n_1}{\hat{\sigma}^2}\mathrm{Rel}^{\mathrm{Train}}_{\mathrm{LOCO}}(Z) = \frac{\hat{\beta}_z^2}{\hat{\sigma}^2/(n_1\hat{\sigma}^2_{[z]n_1})} = F_z = (t_{\beta_z})^2.$$

- If the relevance by LOCO is evaluated in the test sample, then

$$\mathrm{Rel}_{\mathrm{LOCO}}(Z) = \hat{\beta}_z^2 \hat{\sigma}^2_{[z]n_1,n_2} \Rightarrow \frac{n_1}{\hat{\sigma}^2}\mathrm{Rel}_{\mathrm{LOCO}}(Z) = F_z \, \frac{\hat{\sigma}^2_{[z]n_1,n_2}}{\hat{\sigma}^2_{[z]n_1}} \approx F_z.$$

- Relevance by a random permutation,

$$\mathrm{Rel}_{\mathrm{RP}}(Z) \approx 2\hat{\beta}_z^2 \widehat{\mathrm{Var}}(Z).$$

- Relevance by a ghost variable,

$$\mathrm{Rel}^{\mathrm{Train}}_{\mathrm{LOCO}}(Z) = \hat{\beta}_z^2 \hat{\sigma}^2_{[z]n_2} \Rightarrow \frac{n_1}{\hat{\sigma}^2}\mathrm{Rel}_{\mathrm{Gh}}(Z) = F_z \, \frac{\hat{\sigma}^2_{[z]n_2}}{\hat{\sigma}^2_{[z]n_1}} \approx F_z.$$

($\hat{\sigma}^2_{[z]n_1}$, $\hat{\sigma}^2_{[z]n_1,n_2}$ and $\hat{\sigma}^2_{[z]n_2}$ are consistent estimators of $\mathrm{Var}(\varepsilon_z)$ in $Z = X^T\alpha + \varepsilon_z$)

Summary: LOCO, random permutations, and ghost variables

- It is desirable that any variable relevance method would measure meaningful quantities when it is applied to simple models.
- We have seen that in the multiple linear regression model:
  - LOCO and ghost variables give approximately the same results.
  - The relevance of a variable Z, measured by LOCO or by its ghost variable, is proportional to the classical F statistic used for testing $H_0 : \beta_Z = 0$ against $H_0 : \beta_Z \neq 0$.
  - The relevance of Z measured by random permutations does not reproduce any standard test statistic for the significance of β_Z.
- We conclude that measuring variable relevance by ghost variables combines the advantages of the other two methods:
  - The predictive model has to be fitted only once.
  - In linear regression, it reproduces the significance F statistics.
- When we measure variable relevance by ghost variables in any predictive model, we are in some way extending the concept of variable significance to that model.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Other relevance measures based on perturbations

- Random permutations and ghost variables methods for computing relevance of an explanatory variable Z follow a general scheme:

  > To replace the values of Z in the test set by **"perturbed"** values of them, which are independent of the response variable Y, given the other explanatory variables X.

- Other possibilities of *"perturbation"* of Z have been considered recently in the literature.

Conditional distributions and knockoffs

- Hooker, Mentch, and Zhou (2021) propose to replace z_i by a random value coming from the conditional distribution of $(Z \mid X = x_i)$, which is usually known for simulated data.
- This replacement can be done just once, or it can be repeated several times and then record the average results.
- For realistic settings, where the conditional distribution is unknown, Hooker, Mentch, and Zhou (2021) propose to use the Model-X (MX) knockoff framework proposed by Candès et al. (2018) to generate values of Z.
- In particular, they sample second-order multivariate Gaussian knockoff variables as implemented in the R package `knockoff` (Patterson and Sesia 2022).

Estimated conditional distribution

- In a related (but different) context, feature selection in complex predictive models, Tansey et al. (2022) also deal with the problem of working with an unknown conditional distribution $(Z \mid X = x)$, when they describe the general Holdout Randomized Test (HRT).
- Tansey et al. (2022) model the conditional distribution of $(Z \mid X = x)$ as a mixture of univariate Gaussian distributions.
- They fix the number of components in the mixture at 5.
- Then there are $5 + 5 + (5 - 1) = 14$ conditional parameters to be estimated as functions of the p values of x.
- Tansey et al. (2022) propose to estimate the conditional distribution of $(Z \mid X = x)$ following the proposal of Bishop (1994) on mixture density networks.
- This method uses a neural network with 14 neurons in the output layer (one for each parameter), instead of having just one output neuron as it happens when the goal is to estimate simply the conditional expectation $E(Z \mid X = x)$.

- It is worth to say that the estimation of the conditional distribution models is a complicated task requiring a considerable computing effort.
- On the contrary, ghost variables requires only to estimate the conditional expectation using the regression model preferred by the user.
- For instance, linear or additive models (or their generalized versions, if the nature of X_j requires it) can be used.
- If there are many variables, it may be better to use lasso type estimation.

Example 1. A model with 10 explanatory variables

- We follow an example from Hooker, Mentch, and Zhou (2021).
- A multiple linear regression model with 10 explanatory variables uniformly distributed on $[0,1]$, all independent except perhaps the first two of them, which could be possibly correlated through a Gaussian copula with $\rho = 0$ or $\rho = 0.9$.
- Data are generated from the model

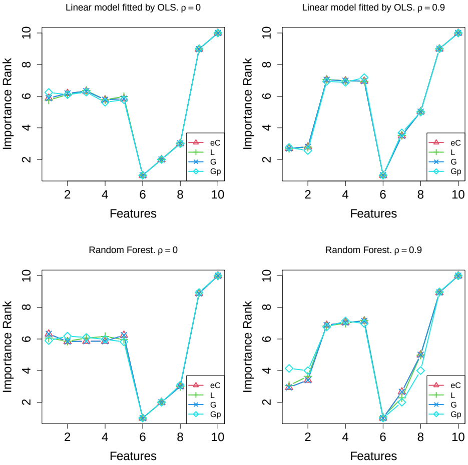
$$Y = x_1 + x_2 + x_3 + x_4 + x_5 + 0x_6 + 0.5x_7 + 0.8x_8 + 1.2x_9 + 1.5x_{10} + \varepsilon,$$

where $\varepsilon \sim N(0, 0.1^2)$. We have repeated 50 times the generation of a training set of size 2000, plus a test set of size 1000.

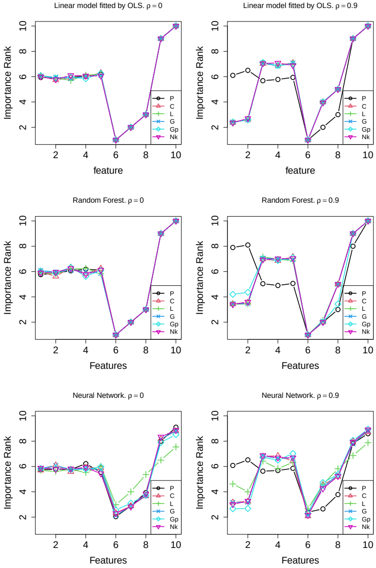
Computation times (in seconds) of different relevance measures applied to several regression models.

Coded in Python	Relevance measures		
			estimated
Models	*loco*	ghost variables	conditional distribution
Linear model (OLS)	0.23	0.34	5102.78
Random forest	1341.62	29.83	12771.48

Coded in R		ghost	true conditional	random	
Lm, RF, NN	*loco*	variables	distribution	permutations	knockoffs
Time (in seconds)	4267.84	49.78	42.93	43.07	46.76



Relative rankings of the explanatory variables according to different relevance measures applied to two regression models. Implementation done in Python.



Relative rankings of the explanatory variables according to different relevance measures applied to three regression models. Implementation done in R.

The main conclusions are the following:

- The random permutation method is giving bad results when there are some inter-dependent features, as expected from our arguments as well as those given by Hooker et al. (2021).
- Ghost variables and knockoffs perform similarly to using random data from the true conditional distributions, with the advantage that the former are feasible in a real setting while the latter is not.
- Ghost variables and knockoffs perform similar to *loco* (except perhaps when fitting neural networks), with the advantage that the former are much faster that the latter.

Knockoffs versus ghost variables

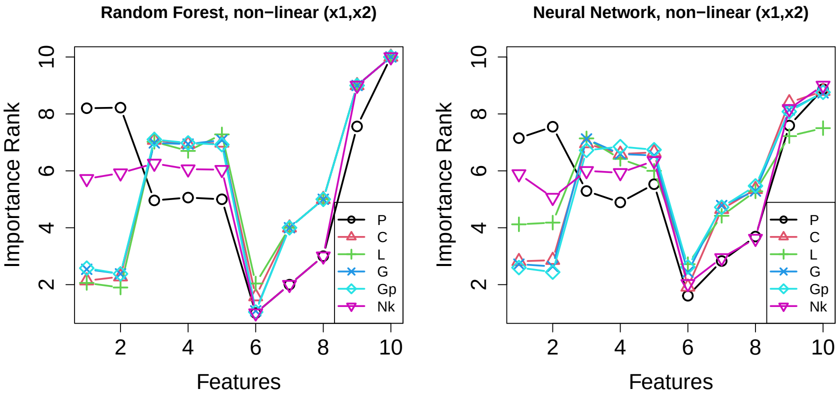
- Using ghost variables or using knockoffs to compute relevance of features are comparable strategies regarding:
  - the quality of the resulting relevance measures,
  - the computational efficiency.
- Both are preferred to other alternatives considered in our simulation study.
- When using ghost variables the practitioner has to propose regression models of each explanatory variable over the others, and then fit these models.
- This is a routine process which is easily implemented in any standard platform (R or Python, for instance), even if the linearity assumption is not fulfilled by our data.
- On the other hand, generating knockoffs variables is difficult even in the most standard settings.
- Moreover, when the data are far from well mimicked with Model-X Gaussian knockoffs there is no easy way to generate knockoffs.
- Simplicity and flexibility of the ghost variable procedure are a clear advantage with respect to using knockoffs.

Nonlinear relation between explanatory variables

- We modify the previous multiple linear regression model introducing a nonlinear dependence between the first two explanatory variables (X_1, X_2).
- First we generate data (θ, R) uniformly in the set $\{[0,\pi/2] \cup [\pi, 3\pi/2]\} \times [0.9, 1]$.
- Then we define

$$X_1 = (R\cos(\theta)+1)/2,\ X_2 = (R\sin(\theta)+1)/2.$$

- This way X_1 and X_2 are both in $[0,1]$ and they present a non-linear dependence pattern.



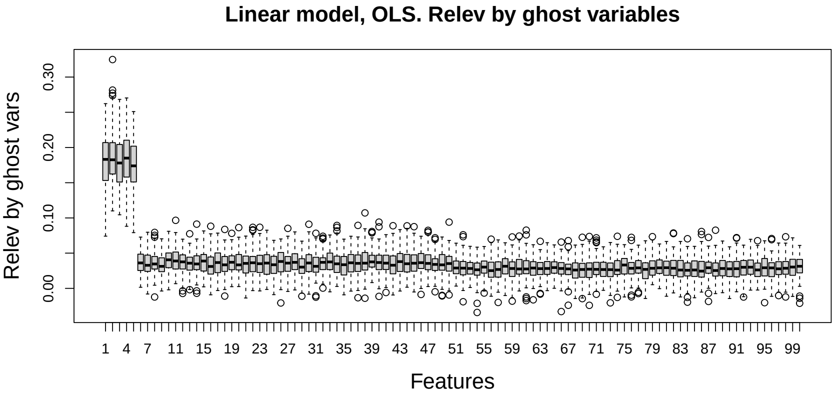
Relative rankings of the explanatory variables according to different relevance measures applied three regression models. A nonlinear dependence pattern has been simulated between the first two explanatory variables.

Example 2. A large model with 100 features

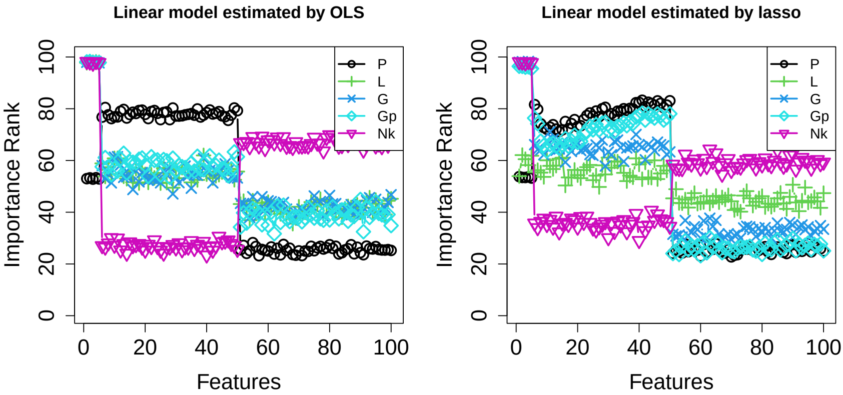
- We simulate now data following a linear model with 100 explanatory Gaussian variables, grouped into three subsets with 5, 45 and 50 variables each, respectively.
- The 5 variables in the first group are independent standard normal.
- In the second group, the 45 variables are marginally standard normal but they are correlated to each other with correlation coefficient $\rho_2 = 0.95$.
- The 50 variables in the third group are independent normal with zero mean and standard deviation $\sigma_3 = 2$.
- Variables in different groups are independent from each other.
- For each observed set of explanatory variables, $x_1, \dots, x_{100}$, the response variable Y is generated from the linear model

$$Y = \sum_{j=1}^{100} \beta_j x_j + \varepsilon,$$

where ε follows a $N(0,1)$ and $\beta_j = \gamma_1 = 0.5$, for $j = 1, \dots, 5$, $\beta_j = \gamma_2 = 1$, for $j = 6, \dots, 50$, and $\beta_j = \gamma_3 = 0.1$, for $j = 51, \dots, 100$.



Relevance of the explanatory variables in the linear model with 100 features of Example 2, estimated by OLS.



- Measuring relevance based on ghost variables gives results at least as good as those of *loco*, with a much lower computational cost.
- Knockoffs is faster than ghost variables in this example, but gives unsatisfactory relevance results.
- The same applies to using random permutations.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

The relevance matrix

- We jointly measure the relevance of all the explanatory variables.
- Given the random vector (X, Y), $X = (X_1, \ldots, X_p) \in \mathbb{R}^p$, $Y \in \mathbb{R}$, the prediction of Y from the p components of X through the estimation of the regression function $m(x) = \mathbb{E}(Y|X = x)$ is considered.
- A training sample $(\mathbf{X}_1, \mathbf{y}_1) \in \mathbb{R}^{n_1 \times (p+1)}$ has been used to build an estimator $\hat{m}(x)$ of $m(x)$.
- An additional test sample $(\mathbf{X}_2, \mathbf{y}_2) \in \mathbb{R}^{n_2 \times (p+1)}$ is available.
- $\mathbf{X}_{2.\hat{j}} = (\mathbf{x}_{2.1}, \ldots, \mathbf{x}_{2.j-1}, \hat{\mathbf{x}}_{2.j}, \mathbf{x}_{2.j+1}, \ldots, \mathbf{x}_{2.p})$, $j = 1, \ldots, p$.
- $\hat{\mathbf{y}}_2 = \hat{m}(\mathbf{X}_2)$, $\hat{\mathbf{y}}_{2.\hat{j}} = \hat{m}(\mathbf{X}_{2.\hat{j}})$.
- Case-variable relevance matrix: $\mathbf{A} = (\hat{\mathbf{y}}_2 - \hat{\mathbf{y}}_{2.\hat{1}}, \ldots, \hat{\mathbf{y}}_2 - \hat{\mathbf{y}}_{2.\hat{p}})$.
- Variable relevance matrix: $\mathbf{V} = \frac{1}{n_2}\mathbf{A}^\mathsf{T}\mathbf{A}$.
- In the diagonal of $\mathbf{V}$: $v_{jj} = \mathrm{Rel}_{\mathrm{Gh}}(X_j)$, $j = 1, \ldots, p$.
- Similarly for random permutations: $\tilde{\mathbf{V}}$.

Comparing relevance matrices in linear regression

Let $\mathbf{S}_2$, $\mathbf{R}$ and $\mathbf{P}$ be, respectively, estimations of the covariance matrix, the correlation matrix, and the partial correlation matrix of X computed in the test sample.

- Ghost variables:

$$\mathbf{V} = -\mathrm{diag}(\hat{\beta})\,\mathrm{diag}(\hat{\sigma}_{[1]},\ldots,\hat{\sigma}_{[p]})\,\mathbf{P}\,\mathrm{diag}(\hat{\sigma}_{[1]},\ldots,\hat{\sigma}_{[p]})\,\mathrm{diag}(\hat{\beta}) =$$

$$\frac{n_2-1}{n_2}\,\mathrm{diag}(\hat{\beta})\,\mathrm{diag}(\hat{\sigma}^2_{[1]},\ldots,\hat{\sigma}^2_{[p]})\,\mathbf{S}_2^{-1}\,\mathrm{diag}(\hat{\sigma}^2_{[1]},\ldots,\hat{\sigma}^2_{[p]})\,\mathrm{diag}(\hat{\beta}).$$

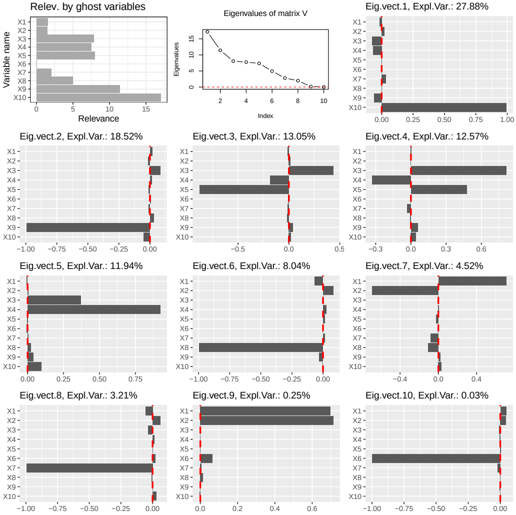
- Random permutations:

$$\tilde{\mathbf{V}} \approx 2\,\mathrm{diag}(\hat{\beta})\,\mathrm{diag}(S_1,\ldots,S_p)\,\mathbf{R}\,\mathrm{diag}(S_1,\ldots,S_p)\,\mathrm{diag}(\hat{\beta}) =$$

$$2\,\mathrm{diag}(\hat{\beta})\,\mathbf{S}_2\,\mathrm{diag}(\hat{\beta}).$$

- In linear regression $\mathbf{V}$ and $\tilde{\mathbf{V}}$ codify complementary information.

Relevance matrix in action: One case from Example 1


Relev. by ghost variables

Eigenvalues of matrix V

Eig.vect.1, Expl.Var.: 27.88%

Eig.vect.2, Expl.Var.: 18.52%

Eig.vect.3, Expl.Var.: 13.05%

Eig.vect.4, Expl.Var.: 12.57%

Eig.vect.5, Expl.Var.: 11.94%

Eig.vect.6, Expl.Var.: 8.04%

Eig.vect.7, Expl.Var.: 4.52%

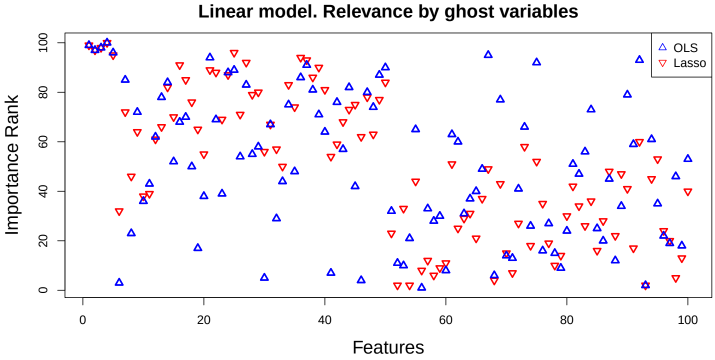
Eig.vect.8, Expl.Var.: 3.21%

Eig.vect.9, Expl.Var.: 0.25%

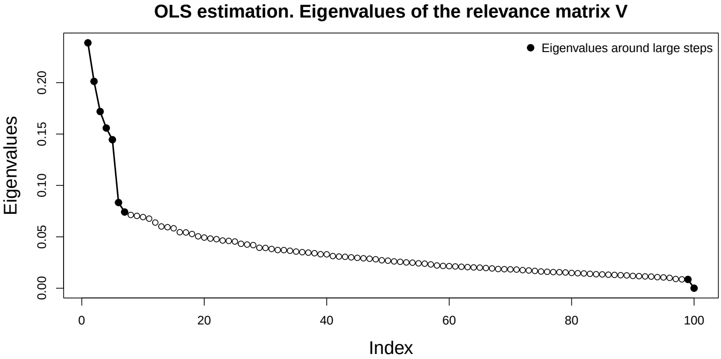
Eig.vect.10, Expl.Var.: 0.03%

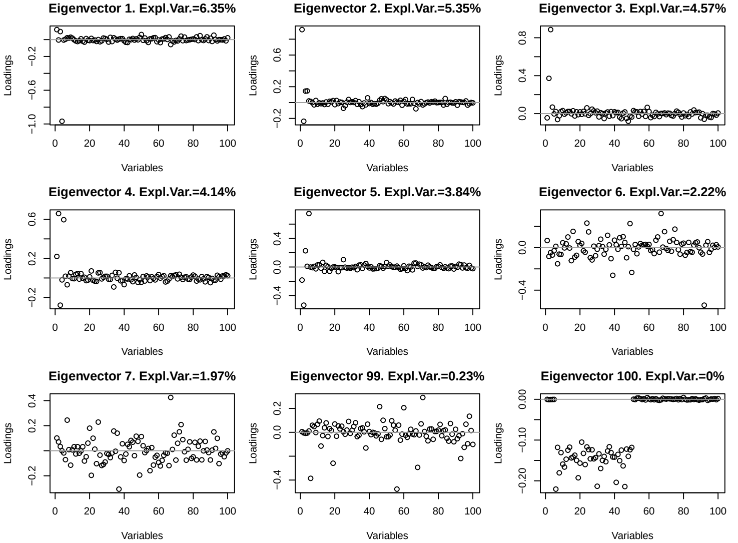
Ghost variables relevance matrix analysis in one data set generated according to Example 1.

One case from Example 1



Ghost variables relevance matrix analysis in one data set generated according to Example 2


OLS estimation. Eigenvalues of the relevance matrix V
● Eigenvalues around large steps
Eigenvalues
0.00
0.05
0.10
0.15
0.20
Index
0
20
40
60
80
100



Cooefficients of each column of **A** in the definition of 9 eigenvectors of **V**.

A real data example: Rent housing prices

- Data on rental housing in Spain, downloaded from Idealista.com on February 27th, 2018, by Alejandro German (Alex seralexger).
- Data available at `https://github.com/seralexger/idealista-data`
- Original data set: 67201 rows (advertisements) and 19 attributes. All cities in Spain.
- We have selected Madrid and Barcelona: 16480 rows.
- Training set 70%, test set 30%.
- Response variable: logarithm of the rental price.
- We work with 16 explanatory variables (some of them calculated from the original data).

```
##  [1] "price"                           "Barcelona"
##  [3] "categ.distr"                     "type.chalet"
##  [5] "type.duplex"                     "type.penthouse"
##  [7] "type.studio"                     "floor"
##  [9] "hasLift"                         "floorLift"
## [11] "size"                            "exterior"
## [13] "rooms"                           "bathrooms"
## [15] "hasParkingSpace"                 "isParkingSpaceIncludedInPrice"
## [17] "log_Days_since_first_activation"
## lm(formula = log(price) ~ ., data = rhBM.price[Itr, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72437 -0.17604 -0.02316  0.15692  1.45330
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      3.8169658  0.0344596 110.766  < 2e-16 ***
## Barcelona                        0.1126307  0.0052554  21.431  < 2e-16 ***
## categ.distr                      0.1169468  0.0033806  34.593  < 2e-16 ***
## type.chalet                     -0.0846942  0.0203106  -4.170 3.07e-05 ***
## type.duplex                     -0.0177992  0.0151519  -1.175  0.24013
## type.penthouse                   0.0428160  0.0101282   4.227 2.38e-05 ***
## type.studio                     -0.0762350  0.0139991  -5.446 5.27e-08 ***
## floor                            0.0128181  0.0009696  13.220  < 2e-16 ***
## hasLift                          0.0480363  0.0118432   4.056 5.02e-05 ***
## floorLift                       -0.0013898  0.0044109  -0.315  0.75270
## log.size                         0.6186668  0.0090654  68.245  < 2e-16 ***
## exterior                        -0.0372539  0.0068935  -5.404 6.64e-08 ***
## rooms                           -0.0501949  0.0034204 -14.675  < 2e-16 ***
## bathrooms                        0.1431973  0.0047167  30.359  < 2e-16 ***
## hasParkingSpace                 -0.0074934  0.0129971  -0.577  0.56426
## isParkingSpaceIncludedInPrice   -0.0408757  0.0138863  -2.944  0.00325 **
## log_Days_since_first_activation  0.0418803  0.0018552  22.574  < 2e-16 ***
## ---
##
## Residual standard error: 0.2647 on 11519 degrees of freedom
## Multiple R-squared:  0.7602, Adjusted R-squared:  0.7599
## F-statistic:  2282 on 16 and 11519 DF,  p-value: < 2.2e-16
```

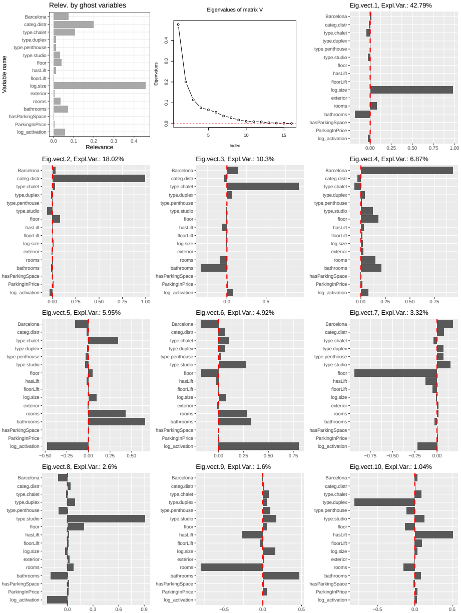
Neutral network fit

- Tuning parameters `size` and `decay` are chosen using `caret`.
- `size` in `c(10,15,20)`, `decay` in `c(0,.1,.3,.5)`.

```
# > nnet.logprice
#
# a 16-10-1 network with 181 weights
#
# inputs: Barcelona categ.distr type.chalet type.duplex type.penthouse type.studio
# floor hasLift floorLift log.size exterior rooms bathrooms hasParkingSpace
# isParkingSpaceIncludedInPrice log_Days_since_first_activation
#
# output(s): .outcome
# options were - linear output units  decay=0.5


# > 1-mean(nnet.logprice$residuals^2)
# [1] 0.8009131
```

The relevance matrix



Rent housing prices: Relevance by ghost variables for a neural network.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Variable relevance measures as Shapley's values

- Consider a linear regression model with response y and explanatory variables $x_1, \dots, x_p$, estimated by OLS from the sample $\{(x_i = (x_{i1}, \dots, x_{ip}), y_i) : i = 1, \dots, n\}$.
- Let $\bar{y} = (1/n) \sum_{i=1}^n y_i$ and let $\hat{y}_i$ be the i-th fitted value.
- A quality measure of the model is the coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}.$$

- Moreover, R^2 is equal to the squared sampling correlation coefficient between the observed responses y_i and the fitted values $\hat{y}_i$.
- When the p explanatory variables are uncorrelated,

$$R^2 = \sum_{j=1}^p R_j^2,$$

  where R_j^2 is the coefficient of determination in the simple linear regression of y against the j-th explanatory variable x_j.
- Therefore, R_j^2 is the contribution of x_j to the global quality measure R^2, and it is a good measure of the relevance of x_j in the model.

- The previous decomposition of R^2 is not longer true when the explanatory variables are correlated.
- Lipovetsky and Conklin (2001) propose an alternative decomposition of R^2, based on Shapley value[1] (Shapley 1953), a useful tool coming from the cooperative games field.
- A cooperative game with p players, $P = \{1, \dots, p\}$, is characterized by a payoff function $v : 2^P \longrightarrow \mathbb{R}$ such that for any coalition of players $S \subseteq P$ the total payoff the members of S can obtain is $v(S)$.
- It is assumed that $v(\emptyset) = 0$.
- When all the players collaborate, the total payoff is $v(P)$.
- The relevant question in cooperative games theory is to find a fair distribution of $v(P)$ among the p players, $\phi_i(v)$, $i = 1, \dots, p$, or rephrased differently, to determine the importance of each player to the overall cooperation.

[1] He won the Nobel Prize in Economic Sciences for this contribution in 2012.

Shapley Value Axiomatic Definition: Four desirable properties

- **Efficiency.** The global payoff is distributed among the players. That is, the sum of the individual payoffs of all players equals the value of the grand coalition, so that all the gain is distributed among the players:

$$\sum_{i \in P} \phi_i(v) = v(P).$$

- **Symmetry.** If i and j are two players who are equivalent in the sense that

$$v(S \cup \{i\}) = v(S \cup \{j\})$$

for every subset S of P which contains neither i nor j, then $\phi_i(v) = \phi_j(v)$. This property is also called "equal treatment of equals".

- **Linearity.** If two coalition games described by payoff functions v and w are combined, then the distributed gains should correspond to the gains derived from v and the gains derived from w:

$$\phi_i(v + w) = \phi_i(v) + \phi_i(w)$$

for every i in P. Also, for any real number a,

$$\phi_i(av) = a\phi_i(v)$$

for every i in P.

- **Null player.** The payoff $\phi_i(v)$ of a null player i in a game v is zero. A player i is "null" in v if $v(S \cup \{i\}) = v(S)$ for all coalitions S that do not contain i.

- **Theorem:** The only distribution satisfying these four desirable properties (axioms) is the Shapley value of the game defined as follows for the j-th player:

$$\phi_j(v) = \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} \left(v(S \cup \{j\}) - v(S)\right).$$

- The quantity $(v(S \cup \{j\}) - v(S))$ is the marginal contribution of player j to the coalition S, and its Shapley value $\phi_j(v)$ is the average of these marginal contributions over the possible different permutations of the set P (S being formed by the elements preceding j at each permutation).
- In fact, an alternative expression for the Shapley value (see, e.g., Cohen, Dror, and Ruppin 2007) is

$$\phi_j(v) = \frac{1}{p!} \sum_{\pi \in \Pi(P)} \left(v(S_j(\pi) \cup \{j\}) - v(S_j(\pi))\right),$$

where $\Pi(P)$ is the set of permutations over P, and $S_j(\pi)$ is the set of players preceding j in permutation π.

- Lipovetsky and Conklin (2001) propose to consider the cooperative game at which the p players are the explanatory variables.
- For a subset S of the p predictors, the characteristic function $v(S)$ is the coefficient of determination R^2_S in the regression of y against the variables belonging to S.
- Therefore, the Shapley value of this game is a fair distribution of the total R^2 among the p predictors: $R^2 = \sum_{j \in P} \phi_j(v)$, and $\phi_j(v)$ measures the importance of the j-th regressor in the model.
- Given that the exact computation of Shapley values is quite time intensive, Lipovetsky and Conklin (2001) suggest to average over a moderate number of random permutation of the explanatory variables.
- This average estimation is founded in the second alternative expression for $\phi_j(v)$ given above, that for linear regression models was previously proposed by Lindeman, Merenda, and Gold (1980) (see also Grömping 2009) with arguments not based on cooperative games.

- Cohen, Dror, and Ruppin (2007) propose to use the Shapley value as global measure of variable relevance in classification problems using any prediction model (or algorithm).
- They propose to use the accuracy in a test set as the characteristic function $v(S)$.
- Note that the calculation of the Shapley value requires fitting the prediction model as many times as different subsets $v(S_j(\pi))$ and $v(S_j(\pi) \cup \{j\})$ are found.
- This task can be prohibitive (even if sampling over permutations is done) for prediction models with moderate or large fitting cost.

Practice:
Washington D.C. Bike Sharing Dataset

Follow the points
2. Shapley Values
3. Relevance by Ghost Variables
in the R-markdown file
`eBISS_IML_bike_sharing_data.Rmd`.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Global graphical methods.

- The general problem of graphically representing a function f depending on p variables, $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$, is not an easy one.
- We present here several approaches that take into account that the function f to be represented is a prediction function estimated from a training set with data assumed to come from a p-dimensional random variable X.
- Availability of an additional test set is also assumed.
- We refer to either the training or the test set as $\{x_i : i = 1, \ldots, n\}$.

Partial Dependence Plot

- The Partial Dependence Plot (PDP), introduced by Friedman (2001) in the context of boosting, is useful for any prediction model.
- The PDP corresponding to the j-th variable aims to represent the j-th partial dependence profile function, defined as

  $$f_j(z) = \mathbb{E}(f(X_{(-j)}, z)) \text{ for all } z \in [x_{\min,j}, x_{\max,j}],$$

  where the notation $(X_{(-j)}, z)$ refers to the p dimensional random vector having all the coordinates as X except the j-th, that is constant and equals to z.
- $[x_{\min,j}, x_{\max,j}]$ denotes the support of the j-th marginal of X.
- The natural estimator of $f_j(z)$ is

  $$\hat{f}_j(z) = (1/n) \sum_{i=1}^{n} f(x_{i(-j)}, z).$$

- Therefore the j-th PDP is the graphical representation of the $\hat{f}_j$.
- For any additive model $f(x) = \alpha_0 + \sum_j g_j(x_j)$, the j-th PDP is the graph of $g_j(z) + C_j$, for some constant C_j.

Local-dependence plots, or marginal plots

- When explanatory variables are not independent, much more interesting than PDPs would be the plots representing the conditional expectation functions

  $$h_j(z) = \mathbb{E}(f(X)|X_j = z)$$

  because the relevant distribution of $X_{(-j)}$ when $X_j = z$ is not the marginal distribution of $X_{(-j)}$ but the conditional distribution $(X_{(-j)}|X_j = z)$.
- The functions $h_j(z)$ can be estimated using any non-parametric regression tool to smooth the scatter plot

  $$(x_{ij}, f(x_{i(-j)}, x_{ij})) = (x_{ij}, f(x_i)), i = 1, \dots, n.$$

- The graphical representation of the estimated functions $h_j(z)$ are known as local-dependence plots (Section 18.3.1 in Biecek and Burzykowski 2021) and as marginal plots (Apley and Zhu 2020).

- Nevertheless, local-dependence plot are not fully satisfactory when there are interactions between explanatory variables in the definition of f, because a problem of omitted variables can appear, as pointed out by Apley and Zhu (2020).
- Consider, for instance, the function $f(x_1, x_2) = x_2 - 0.1x_1x_2$ and (X_1, X_2) uniformly distributed in the set

$$\mathcal{U} = \{|x_1 - x_2| \leq 0.1\} \cap ([0, 1] \times [0, 1]) \, .$$

- The local dependence function corresponding to the first explanatory variable is, for $z \in [0, 1, 0.9]$, $h_1(z) = z - 0.1z^2$, with $h_1'(z) = 1 - 0.2z > 0$ even if $f(x_1, x_2)$ is decreasing in x_1 for any $(x_1, x_2) \in \mathcal{U}$.
- Apley and Zhu (2020) overcome this difficulty by introducing the accumulated local effects (ALE) plots.

Accumulated local effects (ALE) plot

- The ALE plot definition for $p = 2$ explanatory variables is as follows (see Apley and Zhu 2020 for the general definition).
- The local effect of x_1 on f at (x_1, x_2) is computed as the partial derivative $f^1(x_1, x_2) = \partial f(x_1, x_2)/\partial x_1$.
- Therefore,

$$\mathbb{E}\left(f^1(X_1, X_2)|X_1 = x_1\right) = \mathbb{E}\left(f^1(x_1, X_2)|X_1 = x_1\right)$$

  is the conditional expected local effect of x_1 on f.
- Then the accumulated local effect of the first argument of f until the value x_1 is defined as

$$f_{1,ALE}(x_1) = \int_{x_{\min,1}}^{x_1} \mathbb{E}\left(f^1(X_1, X_2)|X_1 = z\right) dz,$$

  where $x_{\min,1}$ is the lower bound of the support of X_1.
- The ALE plot is the graphical representation of $(x_1, f_{1,ALE}(x_1))$ for $x_1 \in [x_{\min,1}, x_{\max,1}]$.

- For a linear function $f(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$, the conditional expected local effect of x_1 at f is always equal to β_1, and the ALE plot is the graph of the straight line $(x_1, \beta_1 x_1 + C)$ for $C = -\beta_1 x_{\min,1}$.
- For an additive model $f(x_1, x_2) = \alpha_0 + g_1(x_1) + g_2(x_2)$, the conditional expected local effect of x_1 at f is equal to $g_1'(x_1)$, and the ALE plot is $(x_1, g(x_1) + C)$ for $C = -g_1(x_{\min,1})$.
- For the previous example of $f(x_1, x_2) = x_2 - 0.1 x_1 x_2$, $f_{1,ALE}(x_1)$ is approximately equal to $-0.05 x_1^2$, correctly reflecting the negative dependence of f with respect to x_1.

- Apley and Zhu (2020) show that the computation cost of ALE plots is lower than that of PDPs.
- Several examples of use of partial-dependence plots, local-dependence plots, and accumulated local effects plots can be found in Chapter 18 of Biecek and Burzykowski (2021).
- These authors point out that the three methods could provide different results when dependence between explanatory variables, and/or interaction effects are present.
- When this is the case, they recommend to explore these causes and take them into account.

Practice:
Washington D.C. Bike Sharing Dataset

Follow the point
4. Global Importance Measures and Plots using the library DALEX
in the R-markdown file
`eBISS_IML_bike_sharing_data.Rmd`.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Local measures

- This section is devoted to model-agnostic methods that provide explanations for a single prediction $f(x)$ of a non-transparent model f.
- Most of the local explanation methods have a common structure: a simple interpretable method g is fitted locally around x in such a way that $g(x') \approx f(x')$ when x' is in a neighborhood of x.
- Then it is expected that the available interpretation of g remains valid for f around x.
- This general approach (called explanation by simplification by Barredo-Arrieta et al. 2020) has certain similarities to local polynomial fitting in non-parametric regression (see, e.g., Fan and Gijbels 1996).

1 Introduction to model-agnostic interpretability methods

2 Global measures of variable relevance
- Leave-one-covariate-out, LOCO
- Variable importance by random permutations
- Relevance by ghost variables
- Other relevance measures based on perturbations
- The relevance matrix
- Variable relevance measures as Shapley's values
- Global graphical methods.

3 Local measures
- LIME
- Explaining individual predictions from Shapley values
- Local graphical methods

4 IML in R and Python

Local Interpretable Model-agnostic Explanations (LIME)

- Among the methods of explanation by simplification, Local Interpretable Model-agnostic Explanations (LIME, Ribeiro, Singh, and Guestrin 2016) is probably the most popular one.
- In LIME, $d << p$ easily recognizable properties of x are selected (e.g., if x is a car image, a property can be the presence of a wheel in the image; if x is a text, a property could be the presence of a certain key word), and their influence in the prediction $f(x)$ is explored.
- The simple interpretable model g is assumed to take values in d-dimensional space, and only $z \in \{0,1\}^d$ are allowed as arguments of g.
- Let $\mathcal{G}$ be the class of models to which g belongs to (for instance, $\mathcal{G}$ can be the class of linear models with d explanatory variables).

- Additionally, a one-to-one application h_x is established between the elements in $\{0,1\}^d$ and 2^d neighbors $x' \in \mathbb{R}^p$ of x.
- For $r = 1, \dots, d$, $z_r = 1$ means that $h_x(z)$ shares the r-th selected property with x. It is assumed that $h_x(1_d) = x$, where 1_d is the vector of ones in $\mathbb{R}^d$.
- The definition of h_x is specific for each problem at hand.
- For instance, when x is a text and the selected properties are the presence of d chosen key words contained in x, $h_x(z)$ returns the text x without the key words for which $z_r = 0$.
- If x is a car image, and the r-th property is the presence of a super-pixel contained in x showing a wheel, for a z with $z_r = 0$ the function $h_x(z)$ will return the same image x with the wheel super-pixel replaced by gray colored pixels.

- A sufficiently large number $N \leq 2^d$ of random points $z \in \{0,1\}^d$ are chosen, and the pairs $(z_k, f(h_x(z_k)))$, $k = 1 \ldots, N$, are annotated.
- The explanation produced by LIME to the prediction $f(x)$ is

$$g_x = \arg\min_{g \in \mathcal{G}} \left\{ \sum_{k=1}^{N} L(g(z_k), f(h_x(z_k)))\pi_x(h_x(z_k)) + \Omega(g) \right\},$$

  where $L(y, y')$ is a loss function measuring how different the real numbers y and y' are, $\pi_x(x')$ is a proximity measure between x and x' (as the kernel function used in non-parametric regression), and Ω is a measure of complexity of the models belonging to $\mathcal{G}$.
- Not every $g \in \mathcal{G}$ may be equally interpretable.
- For instance, for linear models $\Omega(g)$ can be a L_1 penalty term in order to favor sparse solutions, as in LASSO estimation (Tibshirani 1996).

- In practical examples, Ribeiro, Singh, and Guestrin (2016) apply LIME with
    - quadratic loss $L(y, y') = (y - y')^2$,
    - a Gaussian Kernel as $\pi_x(x')$ (with the euclidean distance between x and x', possibly replaced by a more suitable one at each particular case),
    - linear models as family $\mathcal{G}$,
    - LASSO estimation (or least squares after having selected $K < d$ variables by LASSO).
- Using these setting, the LIME explanation g_x for $f(x)$ consists on the selection of K properties among the d that originally were of interest, plus the corresponding estimated coefficients.

- As it was introduced in Ribeiro, Singh, and Guestrin (2016), LIME is more a general methodology than a specific method for local explanation.
- Many details have to be tuned before LIME can be applied to a specific problem.
- Some of them have been already mentioned:
    - class of models,
    - loss function,
    - proximity function (if a kernel function is chosen, the bandwidth is an additional tuning parameter),
    - penalty term,
    - number K of selected properties among the d available.
- There are other important aspects that must also be specified:
    - how many properties d to choose and how they are chosen,
    - how to fix the sample size N,
    - how to define the one-to-one function h_x.

- In fact, Ribeiro, Singh, and Guestrin (2016) do not explicitly talk about the function h_x (they just say that they recover the sample in the original space) that were formalized later in the unifying paper by Lundberg and Lee (2017) when they reviewed LIME (we are coming back to this paper soon).
- In a posterior paper (Ribeiro, Singh, and Guestrin 2018) the authors of LIME focus on the binary classification problem and are much more specific when defining anchors (if-then rules providing high-precision local model-agnostic explanations), a tool that outperform LIME in this context.
- See Section 5.7 in Molnar (2019) and Chapter 9 in Biecek and Burzykowski (2021) for more details on LIME, and examples of applications to image or text data.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

Explaining individual predictions from Shapley values

- Štrumbelj and Kononenko (2010) present a general method for explaining individual predictions of classification models, based on Shapley values.
- Their proposal extends easily to regression problems.
- Given an instance x, the goal is to explain how its feature values $(x_1, \ldots, x_p)$ contribute to the prediction difference between $f(x)$ and the expected prediction if no feature values are known.
- Štrumbelj and Kononenko (2010) consider the framework of a cooperative game at which the p players are the observable features.
- Assuming that the feature values are random observations of a p-dimensional random variable X, the characteristic function proposed by Štrumbelj and Kononenko (2010) is $v_x(\emptyset) = 0$ and, for any nonempty subset S of $P = \{1, \ldots, p\}$,

$$v_x(S) = \mathbb{E}\left(f(X) \mid X_S = x_S\right) - \mathbb{E}\left(f(X)\right),$$

where x_S is the vector containing the coordinates of x with indices in S (similarly for X_S).

- Using the same arguments and notation as in the global relevance problem, the Shapley value of this game for feature j is

$$\phi_j(v_x) = \frac{1}{p!} \sum_{\pi \in \Pi(P)} \left(\mathbb{E}\left(f(X) \mid X_{S_j(\pi)\cup\{j\}} = x_{S_j(\pi)\cup\{j\}} \right) - \right.$$

$$\left. \mathbb{E}\left(f(X) \mid X_{S_j(\pi)} = x_{S_j(\pi)} \right) \right),$$

  and this is the way Štrumbelj and Kononenko (2010) define theoretically the contribution of feature j to the prediction $f(x)$.
- Observe that

$$f(x) = \mathbb{E}\left(f(X)\right) + \sum_{j=1}^{p} \phi_j(v_x)$$

  because the sum of all the Shapley values is equal to $v_x(P) = f(x) - \mathbb{E}\left(f(X)\right)$.
- Therefore $\phi_j(v_x)$ is effectively measuring how the j-th feature of x is contributing to move the prediction from the information-less prediction $\mathbb{E}\left(f(X)\right)$ to the actual prediction $f(x)$.

- In order to give a feasible version of Shapley value, Štrumbelj and Kononenko (2010) propose a sampling approach which:
  - avoids the initial exponential computational time complexity,
  - does not require to repeatedly retrain the classifier.
- To estimate $\mathbb{E}(f(X) \mid X_S = x_S)$, they assume that X is uniform over the product space of the supports of the marginals $X_1, \ldots, X_p$, say $A_1, \ldots, A_p$, that are assumed to be finite or bounded.
- Note that the multivariate uniformity assumption is equivalent to assume that the p features are independent and uniform.

- A simple random sample of size m in the product set $\Pi(P) \times A_1 \times \cdots \times A_p$ is taken.
- Let $\{(\pi_h, u_h = (u_{h1}, \ldots, u_{hp})) : h = 1, \ldots, m\}$, be a sample realization.
- Then, the proposed estimation of $\phi_j(v_x)$ is

$$\hat{\phi}_j(v_x) = \frac{1}{m} \sum_{h=1}^{m} \left\{ f\left(x, u_h, S_j(\pi_h) \cup \{j\}\right) - f\left(x, u_h, S_j(\pi_h)\right) \right\},$$

  where the notation $f(x, u, S)$ means that the function f is evaluated in a point with its r-th coordinate equal to x_r if $r \in S$, and equal to u_r otherwise, for $r = 1, \ldots, p$.
- Štrumbelj and Kononenko (2010) give a rule for choosing the sample size m, once a precision and a significance level have been fixed.

SHAP: A unifying approach

- Lundberg and Lee (2017) propose a method for local explanations, that they call SHAP (SHapley Additive exPlanations), unifying six existing methods, among them
  - LIME (Ribeiro, Singh, and Guestrin 2016),
  - Shapley value based local explanations (Štrumbelj and Kononenko 2010).
- The SHAP framework has several common elements with LIME:
  - a number $d << p$ of properties of x are selected (here they are called simplified input features) for which presence-absence is codified as a vector $z \in \{0, 1\}^d$,
  - a local one-to-one function h_x is defined from $\{0, 1\}^d$ to the neighborhood of x in $\mathbb{R}^d$,
  - an explanation model $g(z)$ is fitted, which in this case is linear:

$$g(z) = \phi_0 + \sum_{r=1}^{d} \phi_r z_r .$$

Instead of estimating the parameters of $g(z)$ using any linear regression model estimator (as it is done in LIME), Lundberg and Lee (2017) propose three desirable properties (Axioms) for this function:

- Local accuracy: The explanation model g matches the original model f for $z = 0_d$ (the vectors of zeros in $\mathbb{R}^d$) and for $z = 1_d$:

$$\phi_0 = g(0_d) = f(h_x(0_d)) \text{ and } f(x) = f(h_x(1_d)) = g(1_d) = \phi_0 + \sum_{r=1}^{d} \phi_r.$$

- Missingness: If the r-th simplified input feature is not present in x (that is, the r-th coordinate of $h_x^{-1}(x)$ is 0), then $\phi_r = 0$.
- Consistency: For any $z \in \{0,1\}^d$ let $z \setminus r$ the vector z with the r-th coordinate replaced by 0. Let f and f' be two prediction models. If

$$f'(h_x(z)) - f'(h_x(z \setminus r)) \geq f(h_x(z)) - f(h_x(z \setminus r))$$

for all inputs $z \in \{0,1\}^d$, then $\phi_r(f', x) \geq \phi_r(f, x)$.

- Lundberg and Lee (2017) prove that the only explanation linear model g verifying these three properties is the one with $\phi_0 = f(h_x(0_d))$ and coefficients ϕ_r, $r = 1, \ldots, d$, equal to the Shapley values of the cooperative game with d players and characteristic function $v_x(S) = f(h_x(z_S))$ for any $S \subseteq \{1, \ldots, d\}$, where z_S is the vector in $\{0, 1\}^d$ having coordinates 1 for the indices in S.
- The exact computation of the coefficients ϕ_r can be done using the general expression for Shapley values given before or, more efficiently, using the previously seen sampling estimation leading to $\hat{\phi}_r$.
- In any case, SHAP results from Lundberg and Lee (2017) coincide with the results derived from the proposals of Štrumbelj and Kononenko (2010).

Kernel SHAP

- Lundberg and Lee (2017) also prove that the SHAP coefficients ϕ_r, $r = 1, \ldots, d$, can be computed much more efficiently using the LIME framework;
  - with quadratic loss,
  - with no penalty ($\Omega(g) = 0$ for any $g \in \mathcal{G}$), and
  - using what they call Shapley kernel as proximity function:

$$\pi_x(h_x(z)) = \frac{d-1}{\binom{d}{|z|}|z|(d-|z|)},$$

  where $|z|$ is the number of non-zero elements of z.
- The Shapley kernel gives infinite weight to $z = 0_d$ and to $z = 1_d$, enforcing the local accuracy property to be fulfilled.
- With these choices, LIME reduces to a weighted least square problem that can be solved efficiently.
- This way of computing SHAP is known as Kernel SHAP.

Break-down plots

- Staniak and Biecek (2018) introduce the break-down plots, as a simplification of Shapley value based proposals of Štrumbelj and Kononenko (2010) and Lundberg and Lee (2017).
- They propose to decompose the difference $f(x) - \mathbb{E}(f(X))$ as the sum of p terms, each accounting for the contribution of one of the p coordinates of x, but their proposal is much more straightforward:

$$f(x) = \mathbb{E}(f(X)) + \sum_{j=1}^{p} \mathbb{E}(f(x_1, \ldots, x_j, X_{j+1}, \ldots, X_p)) -$$

$$\mathbb{E}(f(x_1, \ldots, x_{j-1}, X_j, \ldots, X_p))$$

(where the second term in the sum is just $\mathbb{E}(f(X))$ for $j = 1$).

- Then the break-down contribution of the j-th coordinate of x to the value $f(x)$ is defined as $\psi_j = \mathbb{E}(f(x_1,\ldots,x_j,X_{j+1},\ldots,X_p)) - \mathbb{E}(f(x_1,\ldots,x_{j-1},X_j,\ldots,X_p))$ and can be easily estimated as

$$\hat{\psi}_j = \frac{1}{n}\sum_{i=1}^{n}\left\{f(x_1,\ldots,x_j,x_{i,j+1},\ldots,x_{ip}) - f(x_1,\ldots,x_{j-1},x_{i,j},\ldots,x_{ip})\right\}.$$

- Staniak and Biecek (2018) propose to represent the estimated break-down contributions in a break-down plot, a horizontal bar-plot (or waterfall plot) with indices j as ordinates and the values $\hat{\psi}_j$ as abscissas.

- A clear downside of this proposal is that it depends on the order of the explanatory variables when interactions between them are present in the prediction function (that is, when $f(x)$ is not additive).
- Staniak and Biecek (2018) propose to adopt a greedy strategy and either a step-down or a step-up approach.
- An alternative is to average the $\hat{\psi}_j$ values across all possible orderings, leading to Shapley values as in Štrumbelj and Kononenko (2010).
- A different proposal is given in Gosiewska and Biecek (2020), the break-down plots for interactions, that are able to capture local interactions between explanatory variables and to visualized them by waterfall plots.

1 Introduction to model-agnostic interpretability methods

2 Global measures of variable relevance
- Leave-one-covariate-out, LOCO
- Variable importance by random permutations
- Relevance by ghost variables
- Other relevance measures based on perturbations
- The relevance matrix
- Variable relevance measures as Shapley's values
- Global graphical methods.

3 Local measures
- LIME
- Explaining individual predictions from Shapley values
- Local graphical methods

4 IML in R and Python

Local graphical methods

- Goldstein et al. (2015) introduced the Individual Conditional Expectation (ICE) plot as a refinement of the PDP: the ICE plot shows the relationship between a specific explanatory variable and the response at the individual level, while the PDP does so in an aggregated way.
- Given the prediction function $f(x)$, $x \in \mathbb{R}^p$, the ICE plot corresponding to the i-th observed case $(x_i, y_i) \in \mathbb{R}^{p+1}$ and the j-th explanatory variables is the plot of the function $f_j^{(i)}(z) = f(x_{i(-j)}, z)$ for $z \in [x_{\min,j} x_{\max,j}]$.
- It is usual to mark the point of interest $(x_{ij}, f(x_i))$ on the ICE plot.
- Observe that this ICE plot shows how the prediction for the i-th case is changing when the value of the j-th predictor X_j is changing from the observed value x_{ij} to any other possible value z of X_j, assuming that other things $x_{i(-j)}$ hold constant, or ceteris paribus in Latin.
- This is the reason why Biecek and Burzykowski (2021) call ceteris-paribus profiles to ICE plots.

- From the definitions of PDP and ICE plots, it follows that the partial dependence profile function $\hat{f}_j(z)$ represented by the PDP is the average over all the n data of the individual conditional profiles $f_j^{(i)}(z)$ represented by the ICE plots.
- A useful graphical representation of the prediction function $f(x)$ consists of drawing in gray color the n ICE profiles $f_j^{(i)}(z)$ at the same plot, and then superimpose in black color their average, the PDP.
- The ability of decomposing the global PDP into individual ICE curves is a nice property that is not shared by the ALE plot: there are not individual curves for the ALE plot, as pointed out by Molnar (2019, Chapter 5).

- Biecek and Burzykowski (2021) suggest to complement ICE plots with additional exploratory plots.
- First, for a given ICE profile $f_j^{(i)}(z)$ (or ceteris-paribus profile, using their terminology) Biecek and Burzykowski (2021, Chapter 11) define its oscillation as $\mathbb{E}(|f_j^{(i)}(X_j) - f(x)|)$, and propose how to estimate it.
- In case of a prediction function f with a large number of explanatory variables p, Biecek and Burzykowski (2021) recommend to represent only the ICE plots with the largest oscillation values.
- Additionally, Biecek and Burzykowski (2021, Chapter 12) propose two local-diagnostic plots:
  - The local-fidelity plot, comparing the distribution of neighboring residuals with that of all the residuals,
  - The local-stability plot, representing in a joint graph the ICE plot for the case of interest x_i and those corresponding to neighboring cases.

Practice:
Washington D.C. Bike Sharing Dataset

Follow the point
5. Local explainers with library DALEX
in the R-markdown file
eBISS_IML_bike_sharing_data.Rmd.

1. Introduction to model-agnostic interpretability methods

2. Global measures of variable relevance
   - Leave-one-covariate-out, LOCO
   - Variable importance by random permutations
   - Relevance by ghost variables
   - Other relevance measures based on perturbations
   - The relevance matrix
   - Variable relevance measures as Shapley's values
   - Global graphical methods.

3. Local measures
   - LIME
   - Explaining individual predictions from Shapley values
   - Local graphical methods

4. IML in R and Python

IML in R and Python

- Regarding model-specific methods, the R and Python libraries implementing CART and random forests usually include functions helping the interpretation of the fitted models.
- See, for instance, the R packages `randomForest` (Liaw and Wiener 2002) and `randomForestSRC` (Ishwaran and Kogalur 2021), and the specific functions they include for computing variable importance.
- Also worth mentioning is the R package `randomForestExplainer` (Paluszynska, Biecek, and Jiang 2020), devoted to explain which variables are most important in a random forest.
- In neural networks, the standard implementations are usually less worried about interpretation than in random forests.
- Therefore there are specific packages devoted to provided interpretation to neural network models fitted elsewhere.
- Among them are the following: `validann` (Humphrey et al. 2017), `NeuralNetTools` (Beck 2018), and `NeuralSens` (Portela González et al. 2020).

- With respect to model-agnostic methods, the books Molnar (2019) and Biecek and Burzykowski (2021) are good guides for exploring the different implementations in R and/or Python of the IML/XAI methods we have revised so far.
- At the end of each chapter, both monographs include examples of use for the corresponding methods, and links to R and Python packages implementing them.
- Biecek and Burzykowski (2021) also include code fragments for R and Python that show how to replicate the examples.
- Molnar (2019) uses mainly the R package `iml` (Molnar, Bischl, and Casalicchio 2018), while Biecek and Burzykowski (2021) tend to use the R package `DALEX` (Biecek 2018).

- Alternatively, Maksymiuk, Gosiewska, and Biecek (2020) offer a broad and updated outlook to the available packages in R to perform IML/XAI, covering also the most popular libraries from Python.
- After having presented a taxonomy of methods for model explanations (similar to that in Biecek and Burzykowski 2021), Maksymiuk, Gosiewska, and Biecek (2020) identify and compare 27 packages available in R and six in Python.
- Some examples of application of several packages are included, and a web page with examples of use for all the revised packages is available (`xai-tools.drwhy.ai`).
- The authors compare the packages in several aspects: variety of implemented methods, interoperability, and time of operation.
- From the variety of implemented methods point of view, `DALEX`, `iml` and `flashlight` (Mayer 2021) stand out.

Maksymiuk, Gosiewska, and Biecek (2020) also identify packages that compute automatic standalone reports, as well as those that are able to compare two or more prediction models fitted to the same data:

- `DALEXtra` (Maksymiuk and Biecek 2020),
- `modelDown` (Romaszko, Tatarynowicz, Urbanski, and Biecek 2019),
- `modelStudio` (Baniecki and Biecek 2019).

Apley, D. W. and J. Zhu (2020).
Visualizing the effects of predictor variables in black box supervised learning models.
Journal of the Royal Statistical Society: Series B (Statistical Methodology) 82(4), 1059–1086.

Baniecki, H. and P. Biecek (2019).
modelstudio: Interactive studio with explanations for ml predictive models.
Journal of Open Source Software 4(43), 1798.

Barredo-Arrieta, A., N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. (2020).
Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI.
Information Fusion 58, 82–115.

Beck, M. W. (2018).
Neuralnettools: Visualization and analysis tools for neural networks.
Journal of Statistical Software, Articles 85(11), 1–20.

Biecek, P. (2018).
DALEX: Explainers for complex predictive models in R.
Journal of Machine Learning Research 19(84), 1–5.

Biecek, P. and T. Burzykowski (2021).
Explanatory model analysis: Explore, explain and examine predictive models.
Chapman and Hall/CRC.

Bishop, C. M. (1994).
Mixture density networks.
Aston University.

Breiman, L. (2001).
Statistical modeling: The two cultures.
Statistical Science 16, 199–231.

Candès, E., Y. Fan, L. Janson, and J. Lv (2018).
Panning for gold: 'model-x' knockoffs for high dimensional controlled variable selection.
Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80(3), 551–577.

Cohen, S., G. Dror, and E. Ruppin (2007).
Feature selection via coalitional game theory.
Neural Computation 19(7), 1939–1961.

Delicado, P. and D. Peña (2019).
Understanding complex predictive models with ghost variables.
arXiv:1912.06407.

Fan, J. and I. Gijbels (1996).
Local polynomial modelling and its applications: monographs on statistics and applied probability 66, Volume 66.
CRC Press.

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

Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin (2015).
Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation.
Journal of Computational and Graphical Statistics 24(1), 44–65.

Gosiewska, A. and P. Biecek (2020).
Do not trust additive explanations.
arXiv:1903.11420.

Grömping, U. (2009).
Variable importance assessment in regression: linear regression versus random forest.
The American Statistician 63(4), 308–319.

Hooker, G., L. Mentch, and S. Zhou (2021).
Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance.
Statistics and Computing 31(82).

Humphrey, G., H. Maier, W. Wu, N. Mount, G. Dandy, R. Abrahart, and C. Dawson (2017).
Improved validation framework and r-package for artificial neural network models.
Environmental Modelling and Software 92, 82–106.

Ishwaran, H. and U. Kogalur (2021).
Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC).
R package version 2.10.0.

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

Lindeman, R., P. Merenda, and R. Gold (1980).
Introduction to Bivariate and Multivariate Analysis.
Glenview, IL: Scott, Foresman.

Lipovetsky, S. and M. Conklin (2001).

Analysis of regression in game theory approach.
Applied Stochastic Models in Business and Industry 17(4), 319–330.

Lundberg, S. M. and S.-I. Lee (2017).
A unified approach to interpreting model predictions.
In *Advances in neural information processing systems*, pp. 4765–4774.

Maksymiuk, S. and P. Biecek (2020).
DALEXtra: Extension for 'DALEX' Package.
R package version 2.0.0.

Maksymiuk, S., A. Gosiewska, and P. Biecek (2020).
Landscape of R packages for explainable artificial intelligence.
arXiv:2009.13248.

Mayer, M. (2021).
flashlight: Shed Light on Black Box Machine Learning Models.
R package version 0.7.4.

Molnar, C. (2019).
Interpretable Machine Learning.
Lulu. com.

Molnar, C., B. Bischl, and G. Casalicchio (2018).
iml: An R package for interpretable machine learning.
Journal of Open Source Software 3(26), 786.

Paluszynska, A., P. Biecek, and Y. Jiang (2020).
randomForestExplainer: Explaining and Visualizing Random Forests in Terms of Variable Importance.
R package version 0.10.1.

Patterson, E. and M. Sesia (2022).
knockoff: The Knockoff Filter for Controlled Variable Selection.
R package version 0.3.5.

Portela González, J., A. Muñoz San Roque, and J. Pizarroso Gonzalo (2020).
NeuralSens: Sensitivity Analysis of Neural Networks.
R package version 0.2.2.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016).
Why should I trust you?: Explaining the predictions of any classifier.
In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM.

Ribeiro, M. T., S. Singh, and C. Guestrin (2018).
Anchors: High-precision model-agnostic explanations.
In *AAAI Conference on Artificial Intelligence*, Volume 18, pp. 1527–1535.

Romaszko, K., M. Tatarynowicz, M. Urbanski, and P. Biecek (2019).
modelDown: automated website generator with interpretable documentation for predictive machine learning models.
Journal of Open Source Software 4(38), 1444.

Shapley, L. S. (1953).
A value for n-person games.
Contributions to the Theory of Games 2(28), 307–317.

Staniak, M. and P. Biecek (2018).
Explanations of model predictions with live and breakDown packages.
ArXiv e-prints.

Tansey, W., V. Veitch, H. Zhang, R. Rabadan, and D. M. Blei (2022).
The holdout randomization test for feature selection in black box models.
Journal of Computational and Graphical Statistics 31(1), 151–162.

Tibshirani, R. (1996).
Regression shrinkage and selection via the lasso.
Journal of the Royal Statistical Society: Series B (Methodological) 58(1), 267–288.

Štrumbelj, E. and I. Kononenko (2010).
An efficient explanation of individual classifications using game theory.
Journal of Machine Learning Research 11(Jan), 1–18.