fmeffects: An R Package for Forward Marginal Effects

by Holger Löwe[1], Christian A. Scholbeck[1], Christian Heumann, Bernd Bischl, and Giuseppe Casalicchio

Abstract Forward marginal effects have recently been introduced as a versatile and effective model-agnostic interpretation method particularly suited for non-linear and non-parametric prediction models. They provide comprehensible model explanations of the form: if we change feature values by a pre-specified step size, what is the change in the predicted outcome? We present the R package fmeffects, the first software implementation of the theory surrounding forward marginal effects. The relevant theoretical background, package functionality and handling, as well as the software design and options for future extensions are discussed in this paper.

1 Introduction

A growing number of disciplines are adopting black box machine learning (ML) models to make predictions, including medicine (Rajkomar et al., 2019; Boulesteix et al., 2020), psychology (Dwyer et al., 2018), economics (Mullainathan and Spiess, 2017; Athey and Imbens, 2019), or the earth sciences (Dueben and Bauer, 2018). Although one can often observe a superior predictive performance of black box models (such as neural networks, gradient boosting, random forests, or support vector machines) over intrinsically interpretable models (such as generalized linear or additive models), their lack of transparency or interpretability is considered a major drawback (Breiman, 2001). This has been a major driver in the development of model-agnostic explanation techniques, which are often referred to by the umbrella terms of interpretable ML (Molnar, 2022) or explainable artificial intelligence (Kamath and Liu, 2021).

Marginal effects (MEs) (Williams, 2012) have been a mainstay of model interpretations in many applied fields such as econometrics (Greene, 2019), psychology (McCabe et al., 2022), or medical research (Onukwugha et al., 2015). MEs explain the effect of features on the predicted outcome in terms of derivatives w.r.t. a feature or forward differences in prediction. They are typically averaged to an average marginal effect (AME) for an entire data set, which serves as a global (scalar-valued) feature effect measure (Bartus, 2005). To explain feature effects for non-linear models, Scholbeck et al. (2024) introduced a unified definition of forward marginal effects (FMEs), a non-linearity measure (NLM) for FMEs, and the conditional average marginal effect (cAME). The NLM is an auxiliary model diagnostic to avoid interpreting local changes in prediction as linear effects. The cAME aims to describe the model via regional FME averages for subgroups with similar FMEs, which can, for instance, be found by recursive partitioning (RP). FMEs, therefore, represent a means to explain models on a local, regional, and global level.

Contributions: We present the R package **fmeffects**, the first software implementation of the theory surrounding FMEs, including the NLM and the cAME. The user interface only requires a pre-trained model and an evaluation data set. The package is designed according to modular principles, making it simple to maintain and extend. This paper introduces the relevant theoretical background of FMEs, demonstrates the usage of the package in the context of a practical use case, and explains the software design.

2 Background on forward marginal effects

FMEs can be used for model explanations on the local, regional (also referred to as semi-global), and global level. These differ with respect to the region of the feature space that the explanation refers to. The local level explains a model/prediction for single observations, the regional level for a certain subspace (or subgroups of observations), and the global level for the entire feature space. Increasing the scope of the explanation requires increasing amounts of aggregations of local explanations (see the illustration by Scholbeck et al. (2020) of aggregations of local explanations to global ones for various methods). This can be problematic for non-parametric models where local explanations can be highly heterogeneous due to non-linear effects or interactions.

[1]H. Löwe and C.A. Scholbeck contributed equally.

2.1 Notation

Let $\widehat{f} : \mathcal{X} \to \mathbb{R}$ be the prediction function of a learned model where $\mathcal{X} \subset \mathbb{R}^p$ denotes the feature space. While our definition naturally covers regression models, for classification models, we assume that $\widehat{f}$ returns the score or probability for a predefined class of interest. A subspace of the feature space is denoted by $\mathcal{X}_{[\,]} \subseteq \mathcal{X}$. The random feature vector is denoted by[1] $\boldsymbol{X} = (X_1, \ldots, X_p)$. Observations are denoted by $\boldsymbol{x} = (x_1, \ldots, x_p) \in \mathcal{X}$. A set of feature indices is denoted by $S \subseteq \{1, \ldots, p\}$. We often index (random) vectors as $\boldsymbol{x}_S$ or $\boldsymbol{X}_S$. We denote set complements by $-S = \{1, \ldots, p\} \setminus S$. With slight abuse of notation, we represent the partitioning of a vector into two arbitrary but disjoint groups by $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_{-S})$, regardless of the order of features. For a single feature of interest, the set S is replaced by an integer index j. We usually assume an evaluation data set $\mathcal{D} = \left(\boldsymbol{x}^{(i)}\right)_{i=1}^{n}$, with $\boldsymbol{x}^{(i)} \in \mathcal{X}$, which may consist of both training and test data.

2.2 Forward marginal effects

The FME can be considered a basic, local unit of interpretation. Given an observation $\boldsymbol{x}$, it tells us how the prediction changes if we change a subset of feature values $\boldsymbol{x}_S$ by a vector of step sizes $\boldsymbol{h}_S$.

$$\text{FME}_{\boldsymbol{x},\boldsymbol{h}_S} = \widehat{f}(\boldsymbol{x}_S + \boldsymbol{h}_S, \boldsymbol{x}_{-S}) - \widehat{f}(\boldsymbol{x}) \qquad \text{for continuous features } \boldsymbol{x}_S$$

Scholbeck et al. (2024) introduced an observation-specific categorical FME whose definition is congruent with the FME for continuous features. The categorical FME corresponds to the change in prediction when replacing x_j by the reference category c_j:

$$\text{FME}_{\boldsymbol{x},c_j} = \widehat{f}(c_j, \boldsymbol{x}_{-j}) - \widehat{f}(\boldsymbol{x}) \qquad \text{for categorical } x_j$$

Note that this definition of a categorical ME differs from the one that is typically found in fields like econometrics (Williams, 2012), where we set x_j to a reference category for all observations and then record the change in prediction resulting from changing the reference category to another category.

Furthermore, it is common to globally average MEs to an average marginal effect (AME) to estimate the expected local effect. For FMEs, this corresponds to:

$$\begin{aligned} \text{AME}_{\mathcal{D},\boldsymbol{h}_S} &= \widehat{\mathbb{E}_{\boldsymbol{X}}\left[\text{FME}_{\boldsymbol{X},\boldsymbol{h}_S}\right]} \\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\widehat{f}\left(\boldsymbol{x}_S^{(i)} + \boldsymbol{h}_S, \boldsymbol{x}_{-S}^{(i)}\right) - \widehat{f}\left(\boldsymbol{x}^{(i)}\right)\right] \end{aligned} \tag{1}$$

Note that for categorical feature changes and observations where $x_j = c_j$, the FME equals 0. In the **fmeffects** package, the categorical AME only consists of observations whose observed feature value differs from the selected category. This approach is motivated by our goal to explain the effects of *changing feature values* on the predicted outcome. For instance, in Fig. 11, we demonstrate the effect of rainfall on the daily number of bike rentals in Washington D.C. by switching each non-rainy day's precipitation status to rainfall. Considering all observations, including rainy days, would obfuscate the interpretation we desire from our model. However, it is important to remember that every AME comprises a different set of points.

2.3 Step size selection

The selection of step sizes is determined by contextual and data-related considerations (Scholbeck et al., 2024). First, the FME allows us to investigate the model according to specific research questions. For instance, we might be interested in the effects of a specific change in a patient's body weight on the predicted individual disease risk. Often, we are interested in an interpretable or intuitive step size. In the case of body weight, typically expressed in kilograms, we could use a 1kg change (for instance, instead of 1g) as a natural increment. Without contextual information, we could use a unit change as a reasonable default; or dispersion-based measures such as one standard deviation, percentages of the interquartile range, or the mean/median absolute deviation.

2.4 Non-linearity measure

For continuous features, we can consider $\boldsymbol{x}_S + \boldsymbol{h}_S$ a continuous transition of feature values. The associated change in prediction may be misinterpreted as a linear effect. This is counteracted by the

[1] Bold letters denote vectors.

NLM, which corresponds to a continuous coefficient of determination R^2 between the prediction function and the linear secant that intersects $\boldsymbol{x}$ and $(\boldsymbol{x}_S + \boldsymbol{h}_S, \boldsymbol{x}_{-S})$ (see Fig. 1). The continuous transition through the feature space is first parameterized as a fraction $t \in [0, 1]$ of the multivariate step size $\boldsymbol{h}_S$:

$$\gamma(t) = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} + t \cdot \begin{pmatrix} h_1 \\ \vdots \\ h_s \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad , \quad t \in [0, 1]$$

The value of the linear secant $g_{\boldsymbol{x},\boldsymbol{h}_S}(t)$ corresponds to:

$$g_{\boldsymbol{x},\boldsymbol{h}_S}(t) = \begin{pmatrix} x_1 + t \cdot h_1 \\ \vdots \\ x_s + t \cdot h_s \\ \vdots \\ x_p \\ \widehat{f}(\boldsymbol{x}) + t \cdot \text{FME}_{\boldsymbol{x},\boldsymbol{h}_S} \end{pmatrix}$$

The mean prediction $\widehat{f}_{\text{mean}}$ on the interval $t \in [0, 1]$ is given by:

$$\begin{aligned} \widehat{f}_{\text{mean}} &= \frac{\int_0^1 \widehat{f}(\gamma(t)) \left|\left| \frac{\partial \gamma(t)}{\partial t} \right|\right|_2 dt}{\int_0^1 \left|\left| \frac{\partial \gamma(t)}{\partial t} \right|\right|_2 dt} \\ &= \int_0^1 \widehat{f}(\gamma(t)) \, dt \end{aligned}$$

The NLM compares the squared deviation between the prediction function and the linear secant to the squared deviation between the prediction function and the mean prediction:

$$\text{NLM}_{\boldsymbol{x},\boldsymbol{h}_S} = 1 - \frac{\int_0^1 \left(\widehat{f}(\gamma(t)) - g_{\boldsymbol{x},\boldsymbol{h}_S}(t)\right)^2 \left|\left| \frac{\partial \gamma(t)}{\partial t} \right|\right|_2 dt}{\int_0^1 \left(\widehat{f}(\gamma(t)) - \widehat{f}_{\text{mean}}\right)^2 \left|\left| \frac{\partial \gamma(t)}{\partial t} \right|\right|_2 dt} \quad \in (-\infty, 1]$$

Fig. 2 illustrates the setting for multivariate feature changes. The NLM can be approximated via numerical integration, e.g., via Simpson's rule.

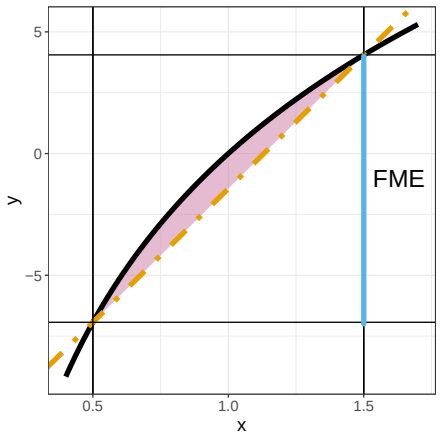

Figure 1: Illustration by Scholbeck et al. (2024) of a univariate FME (blue) given the prediction function (black) and linear secant (orange, dashed). The NLM indicates how well the secant can explain the prediction function (inversely proportional to the purple area) compared to how well the most uninformative baseline model (the average prediction) can explain the prediction function.

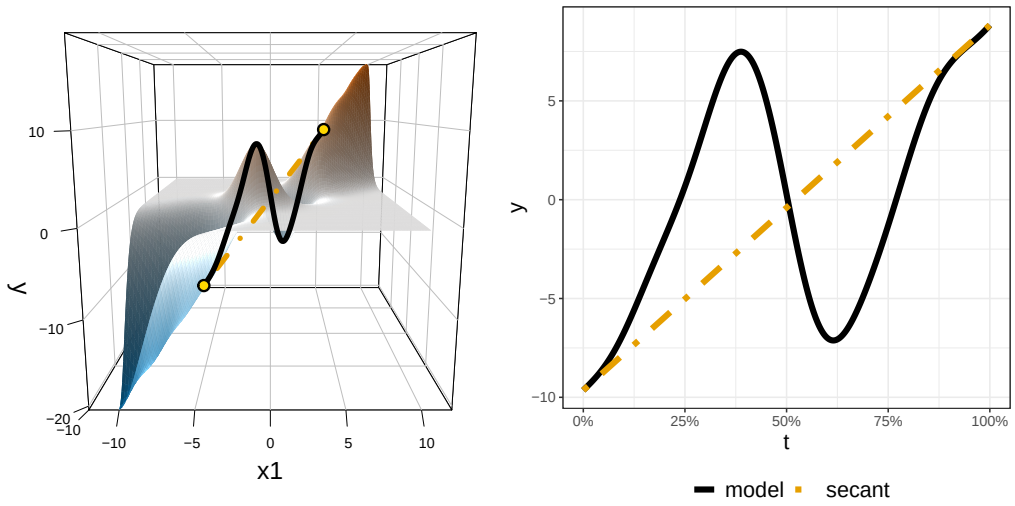

Figure 2: Illustration of the multivariate NLM by Scholbeck et al. (2024). **Left:** An exemplary bivariate prediction function and two points to compute an FME. Consider an observation $\boldsymbol{x} = (-5, -5)$ and step size vector $\boldsymbol{h}_S = (10, 10)$. We create the shortest path through the feature space to reach the point (5, 5), which consists of directly proportional changes in both features. Above the path, we see the linear secant (orange, dashed) and the non-linear prediction function (black). **Right:** The multivariate change in feature values can be parameterized as a percentage t of the step size $\boldsymbol{h}_S$. The deviation between the prediction function and the linear secant, as well as the deviation between the prediction function and mean prediction, both correspond to a line integral.

The NLM indicates how well the linear secant can explain the prediction function, compared to the baseline model of using the mean prediction. A value of 1 indicates perfect linearity, where the linear secant is identical to the prediction function. For a value of 0, the mean prediction can explain the prediction function to the same degree as the secant. For negative values, the mean prediction better explains the prediction function than the linear secant (severe non-linearity).

It is, therefore, easiest to interpret FMEs with NLM values close to 1. Although every FME always represents the exact change in prediction, an FME with a low NLM value does not fully describe the behavior of the model in that specific locality. In contrast, an FME with an NLM close to 1 is a sufficient descriptor of the (linear) model behavior. In other words, the NLM serves as an auxiliary diagnostic tool, indicating trust in how well the FME describes the local change in prediction.

2.5 Conditional average marginal effect

To receive a global model explanation akin to a beta coefficient in linear models, local FMEs can be averaged to the AME. Mehrabi et al. (2021) define an *aggregation bias* as drawing false conclusions about individuals from observing the entire population. Given a data set $\mathcal{D}$, the conditional average marginal effect (cAME) estimator applies to a subgroup of $n_{[\,]}$ observations, denoted by $\mathcal{D}_{[\,]}$:

$$\begin{aligned} \text{cAME}_{\mathcal{D}_{[\,]},\boldsymbol{h}_S} &= \mathbb{E}_{\boldsymbol{X}_{[\,]}}\left[\widehat{\text{FME}_{\boldsymbol{X}_{[\,]},\boldsymbol{h}_S}}\right] \\ &= \frac{1}{n_{[\,]}} \sum_{i:\boldsymbol{x}^{(i)} \in \mathcal{D}_{[\,]}} \left[\widehat{f}\left(\boldsymbol{x}_S^{(i)} + \boldsymbol{h}_S, \boldsymbol{x}_{-S}^{(i)}\right) - \widehat{f}\left(\boldsymbol{x}^{(i)}\right)\right] \end{aligned} \tag{2}$$

Although this estimator can be applied to arbitrary subgroups, we aim to find subgroups with cAMEs that counteract the aggregation bias. Desiderata for such subgroups include within-group effect homogeneity, between-group effect heterogeneity, full segmentation, non-congruence, confidence, and stability (Scholbeck et al., 2024). In other words, we aim to partition the data into subgroups that explain variability in the FMEs. A viable option to partition $\mathcal{D}$ is to run RP on $\mathcal{D}$ with FMEs as the target. For instance, in **fmeffects**, both **rpart** (Therneau and Atkinson, 2019) and `ctree()` from **partykit** (Hothorn and Zeileis, 2015) are supported to find subgroups.

3 Related work

3.1 Model-agnostic interpretations

The basic mechanism behind model-agnostic methods is to probe the model with different feature values, a methodology similar to a model sensitivity analysis (Scholbeck et al., 2020, 2023). The basis of explaining models is to determine the direction and magnitude of the effect of features on the predicted outcome (Casalicchio et al., 2019; Scholbeck et al., 2020, 2024). The individual conditional expectation (ICE) (Goldstein et al., 2015), partial dependence (PD) (Friedman, 2001), accumulated local effects (ALE) (Apley and Zhu, 2020), Shapley values (Štrumbelj and Kononenko, 2010; Lundberg and Lee, 2017; Covert et al., 2020) and local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016) are some of the most popular model-agnostic explanation methods for ML models. Notably, counterfactual explanations (Wachter et al., 2018) represent the reverse of the FME, indicating the smallest necessary change in feature values to reach a targeted prediction.

FMEs complement the literature by allowing for a unique combination of local, regional, and global model explanations. Furthermore, while most methods (including the ICE, PD, ALE, or Shapley values) provide explanations in terms of prediction *levels*, FMEs provide explanations in terms of prediction *changes*. LIME is based on training a local and interpretable surrogate model whose coefficients can also provide an interpretation in terms of prediction changes. Scholbeck et al. (2024) highlighted differences between both approaches: notably, while surrogate models introduce additional uncertainty connected with the estimation of the surrogate, FMEs are motivated by the goal of stable and comprehensible model insight. Furthermore, locally estimated FMEs can be aggregated within subgroups and entire data sets for regional and global explanations. Around the same time, regional aggregations have also been introduced for ICE curves, for example (Britton, 2019; Herbinger et al., 2022; Molnar et al., 2024).

3.2 Relationship between individual conditional expectation and forward marginal effect

Scholbeck et al. (2024) illustrated a relationship between the ICE / PD and the FME / AME. In general, the FME can be seen as the difference between two locations on an ICE. The AME corresponds to the difference between two locations on the PD only for a function that is linear in the feature of interest. Therefore, the following relationship between the ICE and FME is worth noting here. The ICE can be considered a one-way sensitivity function that indicates the effects of varying a set of features indexed by S while the remaining ones are kept constant:

$$\text{ICE}_{\boldsymbol{x},S}(\boldsymbol{x}_S^*) = \widehat{f}(\boldsymbol{x}_S^*, \boldsymbol{x}_{-S})$$

For an instance $\boldsymbol{x}$, the prediction after increasing $\boldsymbol{x}_S$ by $\boldsymbol{h}_S$ is also a value of the ICE:

$$\begin{aligned}\text{FME}_{\boldsymbol{x},\boldsymbol{h}_S} &= \widehat{f}(\boldsymbol{x}_S + \boldsymbol{h}_S, \boldsymbol{x}_{-S}) - \widehat{f}(\boldsymbol{x}) \\ &= \text{ICE}_{\boldsymbol{x},S}(\boldsymbol{x}_S + \boldsymbol{h}_S) - \text{ICE}_{\boldsymbol{x},S}(\boldsymbol{x}_S)\end{aligned}$$

3.3 Related work on marginal effects

MEs have a long history in applied statistics and the Stata programming language (StataCorp, 2023). Initially implemented by Bartus (2005), the `margins()` command is now fully integrated into Stata and provides comprehensive capabilities for various computations and visualizations of statistical models such as (generalized) linear models (Williams, 2012). MEs are typically defined in terms of derivatives of the model w.r.t. a feature. For instance, this variant is the default approach to interpret models in econometrics (Greene, 2019). The FME is the less commonly used definition (Scholbeck et al., 2024; Mize et al., 2019). Note that—in contrast to forward differences—derivatives are not suitable to explain piecewise constant prediction functions such as tree-based models.

In recent years, MEs have gained traction in the R community. The R package **margins** (Leeper, 2018) was the first port of Stata's `margins()` command to R. Other packages related to MEs include **ggeffects** (Lüdecke, 2018) and **marginaleffects** (Arel-Bundock, 2023). In particular, **marginaleffects** can also return FMEs (although under different terminology). Our package, **fmeffects**, mainly differs from **marginaleffects** in two aspects:

Implementing new theory surrounding FMEs: The **fmeffects** package is the first software implementation of the theory surrounding model-agnostic FMEs as introduced by Scholbeck et al. (2024). Although packages such as **marginaleffects** support the computation of FMEs and other quantities, **fmeffects** is specifically designed for FMEs with unique features such as implementations of the NLM, the cAME via RP, and novel visualization methods.

Model-agnostic black box interpretations: It follows that **fmeffects** is targeted at model-agnostic explanations of non-linear or intransparent models. Whereas existing theory on MEs (and packages such as **marginaleffects**) focuses on classical statistical modeling in combination with statistical inference (see, for instance, Breiman (2001) comparing statistical modeling culture with ML), FMEs (and thus **fmeffects**) are comparable to methods and software from the literature on interpretable ML such as the ICE, PD, ALE, or LIME. This does not imply that **marginaleffects** cannot be used for black box interpretations. As mentioned in the previous point, it also supports the computation of FMEs, e.g., in combination with **mlr3**, but the focus of **fmeffects** lies on the interpretation of black box models through a specialized and targeted range of novel capabilities.

4 Advantages and limitations of forward marginal effects

4.1 Advantages

Although the ICE and the FME are closely related, the latter provides several novel ways to generate insights into the model:

- **Univariate changes in feature values:** FMEs are comparable to ICE curves for univariate changes in feature values. In certain scenarios, however, they may provide more comprehensible visualizations of effects for individual instances (see Fig. 4 for an example).
- **Bivariate changes in feature values:** The ICE and PD also provide insight into the sensitivity of the model prediction for variations in two features, which is visualized as a heatmap (see Fig. 7). However, it is difficult to visually compare the ICE of many different observations (which correspond to heatmaps as well). Although the ICE provides insight into a larger variation in feature values, while the FME only considers a single tuple of changes in feature values, bivariate FMEs can be easily compared visually (see Fig. 6).
- **Higher-order changes in feature values:** If we evaluate the sensitivity of the prediction for changes in more than two feature values, virtually every visualization method breaks down. In this case, FMEs still provide comprehensible model explanations that can be aggregated in various ways (see Fig. 10).
- **Local fidelity assessment:** The locally restricted change in feature values for the FME facilitates evaluations of the fidelity of the model explanation (e.g., via the NLM). In other words, the NLM allows us to describe how well the FME summarizes the local shape of the prediction function in a single value. See Fig. 8 for a visualization of NLM values for different observations.
- **Comprehensible regional explanations:** Although regional explanations have been first proposed in the context of grouping ICE curves (Herbinger et al., 2022; Britton, 2019), they more easily apply to scalar model explanations such as FMEs. Essentially, a regional model explanation represents a group of observations or a subspace of the feature space where model explanations are relatively homogeneous. Such groupings are easily achievable via RP or other techniques that do not require functional target values such as ICEs.
- **Avoiding extrapolation:** The ICE is computed on the entire feature range (see, e.g., Fig. 4), which is likely to result in model extrapolations. By its nature, the FME is typically used with small step sizes relative to the feature range, which naturally avoids model extrapolations.

4.2 Limitations

- **Step size selection:** The step size fundamentally influences effects and the model interpretation. Although FMEs for different step sizes can be computed and visualized in an exploratory manner, some level of prior reasoning about what step sizes to use is recommended.
- **Decision tree instability for cAME:** Although not a shortcoming of the FME itself, subgroups found by RP to compute cAMEs are subject to a high variance. This may be counteracted by stabilizing the split search, e.g., by considering statistical significance of tree splits or resorting to different algorithms to find subgroups.
- **Non-linearity assessment for proportional feature changes:** For multi-dimensional feature changes, the NLM only considers equally proportional changes in all features.

5 On causal interpretations and avoiding model extrapolations

Note that model-agnostic techniques, including FMEs, explain associations between the target and the features within the model. In the absence of additional assumptions, such associations cannot be interpreted as causes and effects (Molnar et al., 2022). For instance, increasing the value of a feature x_1 may always be accompanied by an increase in the target, but it may be the target y that causes x_1 to increase. Another typical scenario is the presence of confounding factors that influence both y and x_1. Finally, x_1 may only (or also) influence a mediator x_2, which in turn influences y.

This does not, however, make model interpretations obsolete. More importantly, as highlighted by Adadi and Berrada (2018), model interpretations can be used to gain knowledge, debug, audit, or justify the model and its predictions. Throughout this paper, we will model the effects of environmental influences on the number of daily bike rentals in Washington, D.C. For our estimated model, a drop in humidity by 10 percentage points has a considerable effect on the predicted number of daily bike rentals (see Fig. 5). This effect cannot be assumed to be causal, as humidity is physically influenced by the outside temperature, which will also affect people's choice to rent a bike. Here, temperature is a confounder that influences both humidity and daily bike rentals. However, the business renting out bikes can still use the associations found by a model with a good predictive performance to control the optimal number of bikes at their disposal. This is conditional on the model's ability to accurately predict the target for the given feature vector, requiring us to avoid model extrapolations, which correspond to predictions within areas of the feature space where the model has not seen much or any training data. This issue is closely linked to the multivariate distribution of the training data; in our example, a change in humidity is likely to be accompanied by a change in temperature as well, which we somewhat circumvent (depending on the magnitude of the step size) when making isolated changes to humidity. One may disregard this issue and deliberately predict in areas of the feature space the model has not seen during training. The resulting FMEs will still be valid model descriptions but, as explained above, they are likely to be bad descriptions of the data generating process.

Model extrapolations negatively impact many model-agnostic interpretation methods (Hooker, 2004b,a, 2007; Hooker et al., 2021; Molnar et al., 2022). For example, Apley and Zhu (2020) demonstrated how PD plots suffer from extrapolation issues and introduced ALE plots as a solution to this problem. Scholbeck et al. (2024) illustrated the perils of model extrapolations for FMEs specifically and discussed possible options. One option in particular is also implemented in **fmeffects**: points outside the multivariate envelope (meaning the Cartesian product of all observed feature ranges) of the training data can be excluded from the analysis. This directly relates to the selection of small step sizes relative to the feature range, as large step sizes will result in a point falling outside the envelope.

When using extrapolation prevention methods, note that we consider different sets of points for different step sizes, which differs from the usage of MEs in other contexts (see, for instance, the package **marginaleffects** for a comparison). The exclusion of points only impacts aggregations of FMEs, i.e., the cAME and AME. As discussed in the section on Forward marginal effects, this also affects the computation of categorical AMEs. In Eq. (1) and Eq. (2), the AME and cAME are formulated as estimators of the expected global or regional (concerning a subspace) effects. The fewer observations we are considering for an average, the larger the variance of the estimate.

6 User interface and package handling

6.1 Local explanations

The `fme()` function is the central user interface. It mainly requires a pre-trained model and a data set (see section Design and options for extensions for details). Further control parameters include a list of features and step sizes, whether to compute NLM values for each FME, and an extrapolation detection method. The `fme()` function initiates the construction and computations of a `ForwardMarginalEffect` object without requiring the user to know **R6** (Chang, 2021) syntax.

For this use case, we train a random forest from the **randomForest** package (Liaw and Wiener, 2002) on the bike sharing data set (Fanaee-T, 2013) using **mlr3**. Note that models trained via **tidymodels** and **caret** are also supported, as well as models trained via `lm()`, `glm()`, and `gam()`. We aim to predict and explain the daily bike rental demand in Washington, D.C., based on features such as the outside temperature, wind speed, or humidity. We first train the model:

```
> library(fmeffects)
> data(bikes, package = "fmeffects")
> library(mlr3verse)
> library(mlr3extralearners)
> forest = lrn("regr.randomForest")
```

```
> task = as_task_regr(x = bikes, id = "bikes", target = "count")
> forest$train(task)
```

Then, we simply pass the trained model, evaluation data, and remaining parameters to the `fme()` function. It returns a `ForwardMarginalEffect` object, which can be analyzed via `summary()` and visualized via `plot()` (see Fig. 3). Here, the outside temperature is raised by 5 degrees Celsius ceteris paribus. To avoid overplotting values, each hexagon represents a local average of FMEs. Users can easily access the data used by all plot functions to implement their own visualizations.

Let us single out the observation with the largest associated FME. This observation corresponds to a single day with a recorded temperature of 8 degrees Celsius. Increasing the temperature by 5 degrees Celsius on this particular day results in 2563 additional predicted bike rentals. We plot such model explanations for the entire data set and average FMEs to receive a global model explanation. The AME—the global average of FMEs—is 304: an increase in temperature by 5 degrees Celsius results in an average increase of 304 predicted daily bike rentals.

```
> effects.univariate.temp = fme(
+   model = forest,
+   data = bikes,
+   features = list("temp" = 5),
+   ep.method = "envelope")

> summary(effects.univariate.temp)

Forward Marginal Effects Object

Step type:
  numerical

Features & step lengths:
  temp, 5

Extrapolation point detection:
  envelope, EPs: 48 of 731 obs. (7 %)

Average Marginal Effect (AME):
  304.1722

> plot(effects.univariate.temp)
```

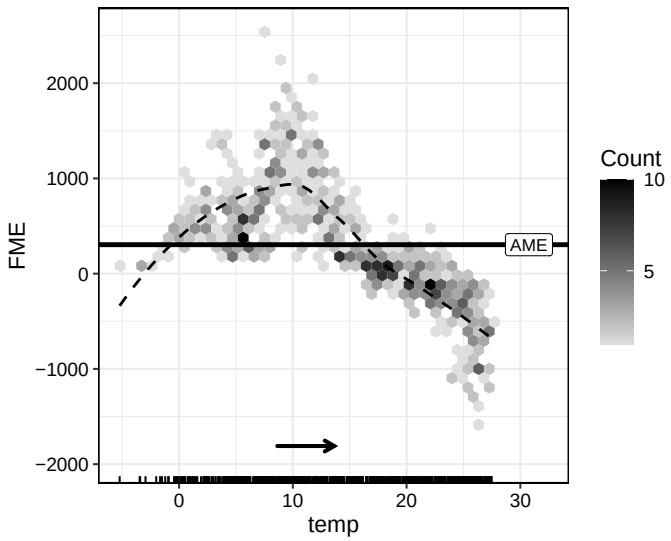

Figure 3: Plot of univariate FMEs for feature '`temp`' and step size 5. Each hexagon represents a local FME average. The horizontal value represents the observed feature value of '`temp`'. Each observation's '`temp`' value is moved according to the arrow's direction and length. The vertical value of each hexagon indicates the FME value associated with that feature change. The horizontal bar indicates the AME. The shade of the hexagon implies how many observations it contains. A smoothing function facilitates interpretations by modeling an approximate pattern of FMEs across the feature range.

Let us take a moment to compare the FME plot with the combined ICE and PD plot generated by the R package **iml** (Molnar et al., 2018) (see Fig. 4). This is one of the most popular and established model-agnostic ways to interpret predictive models (Molnar, 2022). The ICE is a local model explanation and represents the prediction for an observation where only the features of interest are varied (in this case, only 'temp'). The PD is the average of ICEs (in the univariate case, the vertical average) and indicates the global, average prediction when a subset of features is varied for all observations. Although we can see a rough trajectory of the feature influence on local and average predictions, it is difficult to pinpoint the exact effects of changing 'temp' on the prediction for single observations. Furthermore, ICE curves are more likely to be subject to model extrapolations, a result of predicting in areas where the model was not trained on a sufficient amount of data.

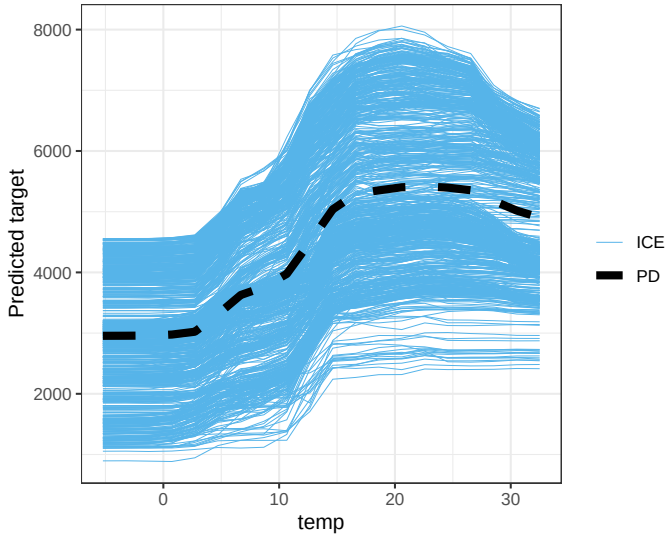

Figure 4: An ICE and PD plot for feature 'temp' generated by the R package **iml**. Each solid blue curve (an ICE) represents predictions for a single instance while only 'temp' varies. The dashed black curve (the PD) is the vertical average of ICEs and represents the average, isolated influence of 'temp'.

FMEs allow for positive or negative step sizes. For instance, let us investigate the effects of an isolated drop in humidity by 10 percentage points. We can observe an AME of 103 additional predicted bike rentals a day. Individual effects tend to be larger the higher the humidity on that particular day.

```
> effects.univariate.humidity = fme(
+   model = forest,
+   data = bikes,
+   features = list("humidity" = -0.1),
+   ep.method = "envelope")

> summary(effects.univariate.humidity)

Forward Marginal Effects Object

Step type:
  numerical

Features & step lengths:
  humidity, -0.1

Extrapolation point detection:
  envelope, EPs: 1 of 731 obs. (0 %)

Average Marginal Effect (AME):
  102.9158

> plot(effects.univariate.humidity)
```

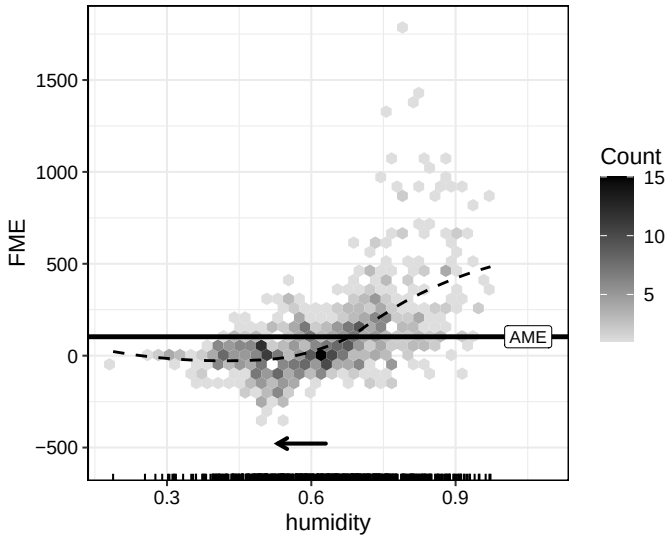

Figure 5: Univariate FMEs for a drop in humidity by 10 percentage points. Especially for high humidity values, the drop results in a considerable increase in predicted daily bike rentals.

In many applications, we are interested in interactions of features on the prediction. Until now, we only analyzed the univariate effects of 'temp' and 'humidity' on the predicted amount of bike rentals. However, potential interactions between features may exist. We evaluate an increase in temperature by 5 degrees Celsius and a simultaneous drop in humidity by 10 percentage points (see Fig. 6). For a bivariate change in feature values, the two arrows depict the direction and magnitude of the feature change in the respective variable. As in the univariate case, we plot local averages within hexagons to avoid overplotting values. The location of the hexagon is determined by the observations' observed feature values in the provided data set. Its color indicates the FME associated with the bivariate feature change. An increase in the outside temperature by 5 degrees Celsius and a simultaneous drop in humidity by 10 percentage points is associated with an AME of 403. The univariate AMEs roughly add up to the bivariate AME, indicating that, on average, there is no additional interaction between both feature changes on the prediction.

```
> effects.bivariate = fme(
+   model = forest,
+   data = bikes,
+   features = list("temp" = 5, "humidity" = -0.1),
+   ep.method = "envelope")

> summary(effects.bivariate)

Forward Marginal Effects Object

Step type:
  numerical

Features & step lengths:
  temp, 5
  humidity, -0.1

Extrapolation point detection:
  envelope, EPs: 49 of 731 obs. (7 %)

Average Marginal Effect (AME):
  403.0714

> plot(effects.bivariate)
```

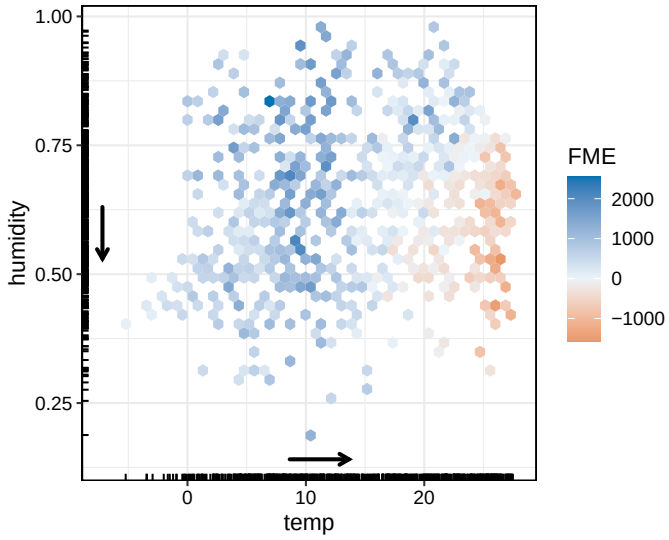

Figure 6: Visualizing bivariate FMEs for an increase in 'temp' by 5 degrees Celsius and a simultaneous drop in 'humidity' by 10 percentage points. FMEs are highly heterogeneous. We can see mostly positive effects, especially for observations with combinations of medium 'temp' and 'humidity' values.

Let us repeat the same procedure as for univariate feature changes and compare the FME plot to an alternative option, the bivariate PD plot (see Fig. 7). As opposed to the novel visualization with FMEs, the PD plot only visualizes the average, global effect of changing both features on the predicted amount of bike rentals. It does not inform us about the distribution of observed feature values, thus not allowing us to evaluate the effects of increasing one feature and decreasing another simultaneously.

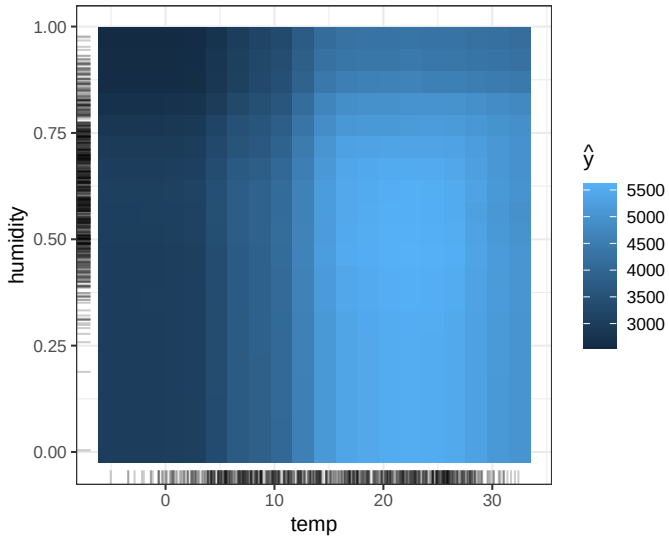

Figure 7: A bivariate PD plot (created via the R package **iml**), visualizing the global interaction between 'temp' and 'humidity' on the predicted amount of bike rentals. Plugging in medium to large values for 'temp' and low to medium values for 'humidity', ceteris paribus, results in more predicted bike rentals on average. As opposed to bivariate FMEs, we cannot investigate multiple local effects, nor can we see the actual distribution of observed feature values. As a result, we cannot evaluate the effects of increasing one feature and decreasing another simultaneously.

Let us now proceed to investigate non-linearity. Non-linearity can be visually assessed for ICE curves (see Fig. 4), but it is hard to quantify and would be somewhat meaningless for a large variation in the feature of interest. Furthermore, for bivariate or higher-dimensional changes in feature values, we lose any option for visual diagnoses of non-linearity. In contrast, the NLM can be computed for FMEs with continuous step sizes, regardless of dimensionality. The average non-linearity measure (ANLM) is 0.34, indicating that the linear secant, on average, is a bad descriptor of the FME.

```
> effects.bivariate.nlm = fme(
+   model = forest,
+   data = bikes,
+   features = list("temp" = 5, "humidity" = -0.1),
+   ep.method = "envelope",
+   compute.nlm = TRUE)

> effects.bivariate.nlm

Forward Marginal Effects Object

Features & step lengths:
  temp, 5
  humidity, -0.1

Average Marginal Effect (AME):
  403.0714

Average Non-Linearity Measure (ANLM):
  0.34

> plot(effects.bivariate.nlm, with.nlm = TRUE)
```

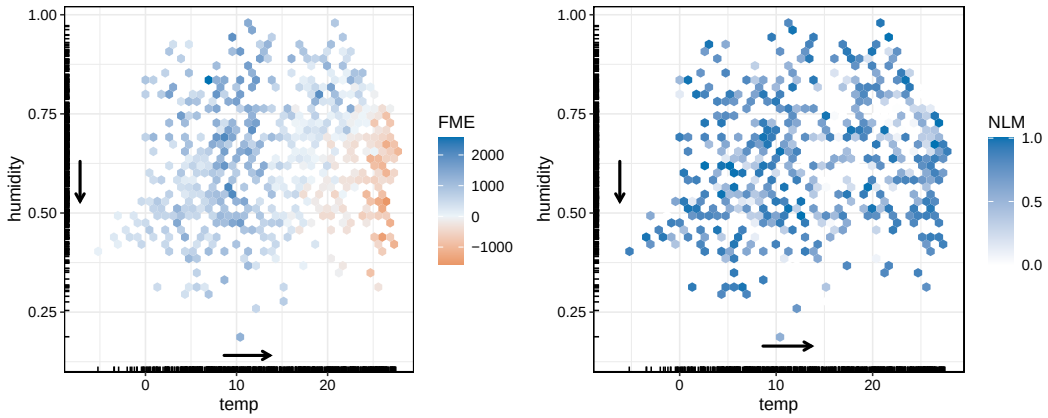

Figure 8: Adding NLM computations to the FME plot. Each hexagon in the left and right plots represents a local average of FME and NLM values, respectively.

Fig. 8 simply contrasts FME values with the corresponding NLM values. In this case, we can see both non-linear FMEs (whiter NLM) and linear FMEs (deep blue-colored NLM). We could now, for instance, focus on interpreting linear FMEs. All FMEs depicted in Fig. 9 have an NLM of 0.9 or higher, meaning that they almost fully describe the model prediction for proportional changes in 'temp' and 'humidity'.

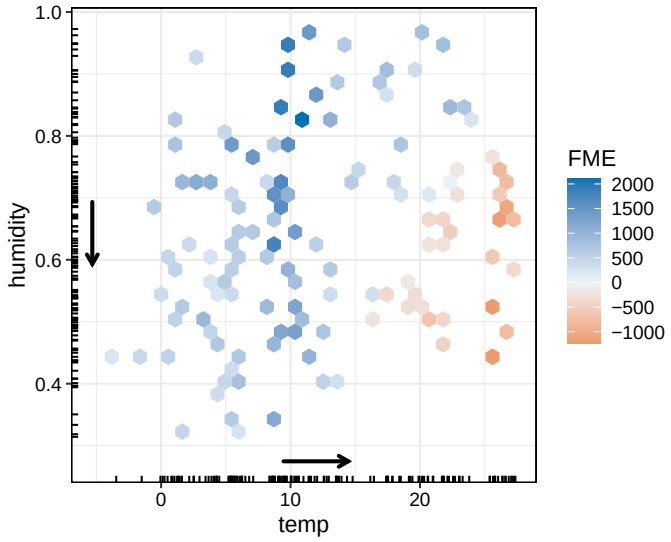

Figure 9: Visualizing FMEs with an NLM ≥ 0.9.

An advantage of FMEs is their ability to provide comprehensible model insight even when exploring higher-order feature changes. Let us factor in a third feature change, now simultaneously reducing windspeed by 5 miles per hour, and visualize the distribution of FME and NLM values. We can see that in addition to an increase in temperature and a decrease in humidity, a decrease in windspeed further boosts the average number of predicted daily bike rentals.

```
> effects.trivariate.nlm = fme(
+   model = forest,
+   data = bikes,
+   features = list("temp" = 5, "humidity" = -0.1, "windspeed" = -5),
+   ep.method = "envelope",
+   compute.nlm = TRUE)

> summary(effects.trivariate.nlm)

Forward Marginal Effects Object

Step type:
  numerical

Features & step lengths:
  temp, 5
  humidity, -0.1
  windspeed, -5

Extrapolation point detection:
  envelope, EPs: 117 of 731 obs. (16 %)

Average Marginal Effect (AME):
  515.2608

Average Non-Linearity Measure (ANLM):
  0.31

> plot(effects.trivariate.nlm, with.nlm = TRUE)
```

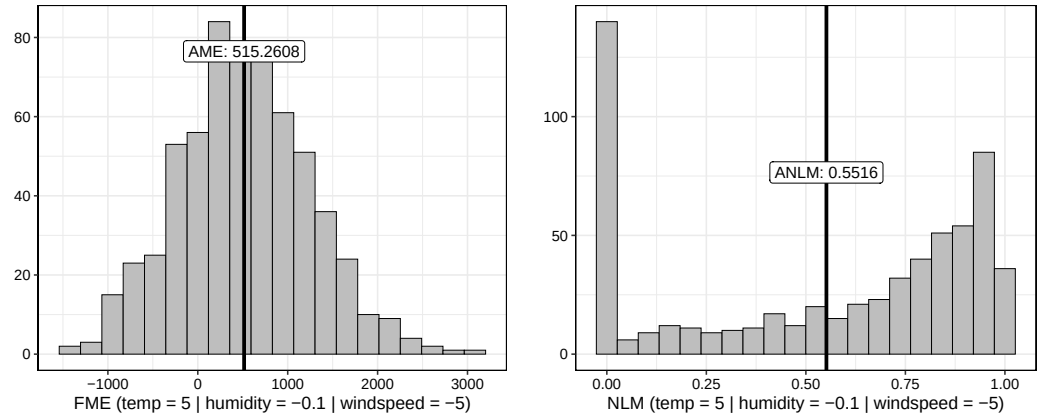

Figure 10: Adding a third feature change, a drop in windspeed by 5 miles per hour, and visualizing the distribution of FME and NLM values. For the NLM plot, negative NLMs are binned as 0. It follows that the ANLM value in the plot differs from the raw ANLM in the summary output.

So far, we have only evaluated changes in continuous features. In many applications, we are concerned with switching categories of categorical features, a way of counterfactual thinking inherent to the human thought process. Note that despite the allure of switching categories of categorical features, one needs to be aware of potential model extrapolations. To illustrate this, we switch each non-rainy day's precipitation status to rainfall. Rainfall has an average isolated effect of lowering daily rentals by 699 bikes (see Fig. 11).

```
> effects.categ = fme(
+   model = forest,
+   data = bikes,
+   features = list("weather" = "rain"))

> summary(effects.categ)

Forward Marginal Effects Object

Step type:
  categorical

Feature & reference category:
  weather, rain

Extrapolation point detection:
  none, EPs: 0 of 710 obs. (0 %)

Average Marginal Effect (AME):
  -699.4915

> plot(effects.categ)
```

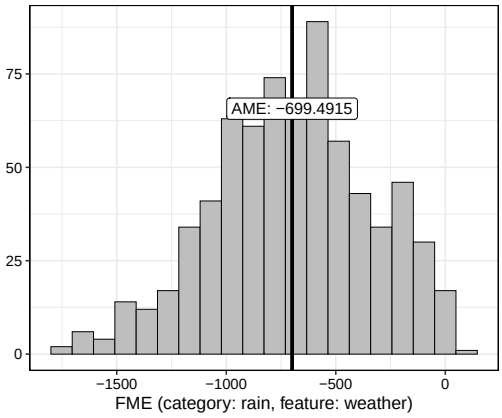

Figure 11: Distribution of categorical FMEs resulting from switching each non-rainy day's precipitation status to rain. On average, rainfall lowers predicted bike rentals by 699 bikes per day.

6.2 Regional explanations

In our examples, we can see highly heterogeneous local effects. The more heterogeneous FMEs are, the less information the AME carries. In many practical applications, we are interested in compactly describing the behavior of the predictive model across the feature space, akin to a beta coefficient in a linear model. This is where regional explanations come into play. We aim to find subgroups with more homogeneous FME values, thereby describing the behavior of the model not in terms of a global average but in terms of multiple regional averages (cAMEs).

In **fmeffects**, this can be achieved by further processing the `ForwardMarginalEffect` object containing FMEs (and optionally NLM values) using the `came()` function. This returns a `Partitioning` object (in this case, an object of the class `"PartitioningCTREE"`, a subclass of the abstract class `"Partitioning"`, see later section on Design and options for extensions).

For the univariate change in temperature by 5 degrees Celsius, we decide to search for precisely 2 subgroups[2] (for a description of this algorithm, see the following section on Design and options for extensions). A summary of the created object informs us about the number of observations, cAME, and standard deviation (SD) of FMEs inside the root node and leaf nodes (the found subgroups). We succeeded in finding subgroups with lower SDs while maintaining an appropriate sample size. The root node SD of 620 can be successfully split down to 442 and 369 within the subgroups. By visualizing the tree, we can see how the data was partitioned. For cooler outside temperatures equal to or lower than ≈ 16 degrees Celsius, we can observe a positive cAME of 730 additional bike rentals per day. On warmer days with a temperature above ≈ 16 degrees Celsius, the model predicts 205 less bike rentals a day when the outside temperature increases by 5 degrees.

```
> subspaces = came(effects = effects.univariate.temp, number.partitions = 2)
> summary(subspaces)

PartitioningCtree of an FME object

Method:  partitions = 2

   n      cAME  SD(fME)
 683  304.1722 620.4775 *
 372  729.8519 441.6201
 311 -205.0011 368.8368
---
* root node (non-partitioned)

AME (Global): 304.1722

> plot(subspaces)
```

[2]This value is to be set by the user depending on how many regional explanations are to be found. Alternatively, we can search for a pre-defined SD of FMEs inside the terminal nodes. How many subgroups can be found depends on the data and predictive model.

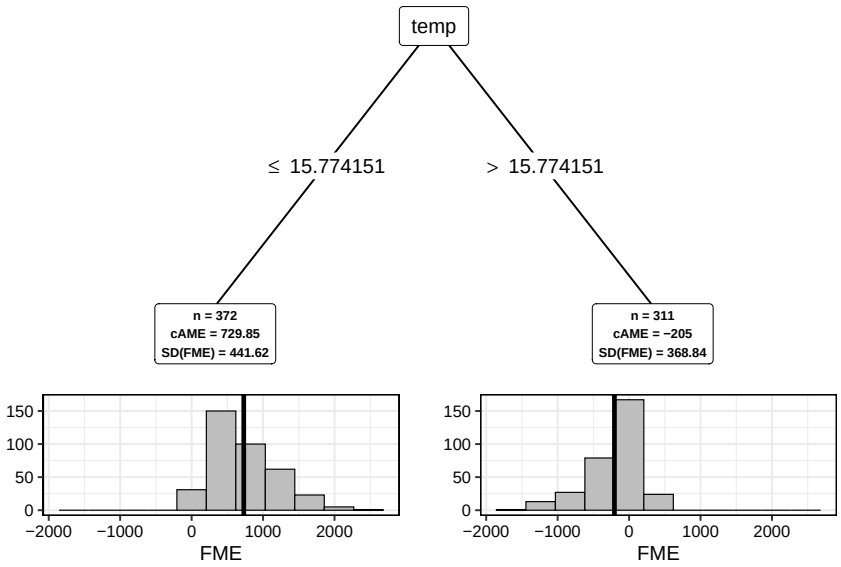

Figure 12: Using a decision tree to find subgroups of observations with more homogeneous FMEs of increasing 'temp' by 5 degrees Celsius. Each leaf node visualizes one subgroup, the number of observations, the cAME, and the SD of FMEs indicating FME homogeneity.

6.3 Global explanations

When to search for regional explanations thus depends on the heterogeneity of local effects. The `ame()` function provides an appropriate summary for the entire model. It uses a default step size of 1 or 0.01 for small feature ranges. For categorical FMEs, it uses every observed category as a reference category. Alternatively, custom step sizes and subsets of features can be used. The `summary()` function prints a compact model summary of each feature, a default step size, the AME, the SD of FMEs, 25% and 75% quantiles of FMEs, as well as the number of observations left after excluding extrapolation points (EPs). A large dispersion indicates heterogeneity of FMEs and thus a small fidelity of the AME and possible benefits from searching for subgroups with varying cAMEs. A different workflow can, therefore, also consist of starting with the table generated by `ame()` and deciding which feature effects can be described by AMEs and which might be better describable by subgroups and cAMEs. If this has been unsuccessful, we can resort to local model explanations. Recall our example from the previous section on Regional explanations where we split FMEs associated with increasing temperature by 5 degrees Celsius. From the `ame()` summary, we see that 'temp' has a relatively large SD in relation to its AME (here calculated with a step size of 1), and the interquartile range indicates a wide spread of FMEs from -21 in the 25% quantile up to 107 in the 75% quantile, which makes it a promising candidate to find subgroups with more homogeneous FMEs.

```
> ame.results = ame(model = forest, data = bikes)
> summary(ame.results)

Model Summary Using Average Marginal Effects:

      Feature step.size        AME       SD       0.25       0.75   n
1      season    winter  -894.4673 456.3625 -1248.2476  -586.5656 550
2      season    spring   141.6627 557.8672  -242.9194   652.8917 547
3      season    summer   538.4263 627.8606    45.0598  1196.3612 543
4      season      fall   493.7475 581.3166     8.7096  1101.5087 553
5        year         0 -1890.8318 641.3168 -2377.7961 -1496.2576 366
6        year         1   1785.563 508.6759  1412.6724  2183.8292 365
7     holiday        no   165.2367 213.3036    72.5637   194.7954  21
8     holiday       yes  -122.4971 141.9902  -189.0043   -22.1315 710
9     weekday    Sunday   107.3675 199.4931   -33.8124   218.4856 626
10    weekday    Monday  -127.8842  232.482   -260.735    23.9211 626
11    weekday   Tuesday  -110.9437 219.9664  -216.2248    29.4189 626
```

```
12    weekday Wednesday   -16.5913 204.4574  -113.3341   118.8563 627
13    weekday  Thursday    27.4835 189.9021   -85.0117   140.1993 627
14    weekday    Friday     53.982 194.2184   -65.3866   170.0411 627
15    weekday  Saturday   110.8837 191.1073    -7.7049   231.8014 627
16 workingday        no    -41.222 115.1556  -126.4856    45.9121 500
17 workingday       yes    42.5305 154.5266   -67.1033   134.7876 231
18    weather     misty  -236.5115 327.3365   -442.211   -71.8195 484
19    weather     clear   368.2611 325.1541   145.7027   459.1031 268
20    weather      rain  -699.4915 362.8458  -943.5127  -454.9041 710
21       temp         1    56.6478 167.5781   -21.1847   106.6103 731
22   humidity      0.01   -20.3705  58.2372   -35.0143      8.289 731
23  windspeed         1   -24.3256  73.3227   -50.7023    12.0791 731
```

7 Design and options for extensions

The **fmeffects** package is built on a modular design for improved maintainability and future extensions. Fig. 13 provides a visual overview of the core design. The greatest emphasis is placed on the strategy and adapter design patterns (Gamma et al., 1994). Simply put, the strategy pattern decouples the source code for algorithm selection at runtime into separate classes. We repeatedly implement this pattern throughout the package by creating abstract classes whose subclasses implement specific functionalities. The adapter design pattern (also called a "wrapper") creates an interface for communication between two classes.

- `"Predictor"`: An abstract class that implements the adapter pattern to accommodate future implementations of storing a predictive model. `"PredictorMLR3"`, `"PredictorParsnip"`, and `"PredictorCaret"` are subclasses that store an **mlr3**, **parsnip** (Kuhn and Vaughan, 2023) (part of **tidymodels**), or **caret** model object. This allows users of **fmeffects** to use numerous predictive models such as random forests, gradient boosting, support vector machines, or neural networks. `"PredictorLM"` stores models returned by `lm()`,`glm()`,or `gam()`. The package can be extended with novel model types by implementing a new subclass that stores the model, data, target, and is able to return predictions.
- `"AverageMarginalEffects"`: A class to compute AMEs for each feature in the data (or a subset of features). Internally, a new `"ForwardMarginalEffect"` object is used to compute and aggregate FMEs. For convenience, we implement a wrapper function `ame()` to facilitate object creation and to initiate computations without requiring user input in the form of **R6** syntax.
- `"ForwardMarginalEffect"`: The centerpiece class of the package. It keeps access to a `Predictor`, stores important information to create FMEs, and after the computations are completed, stores results and gives access to visualization methods. For convenience, the wrapper function `fme()` can be used.
- `"FMEPlot"`: An abstract class for code decoupling of different plot categories into distinct classes. Subclasses include `"FMEPlotUnivariate"`, `"FMEPlotBivariate"`, `"FMEPlotHigherOrder"`, `"FMEPlotCategorical"`.
- `"ExtrapolationDetector"`: Identifies (and excludes) EPs. The current implementation supports the method "envelope", excluding points outside the multivariate envelope of the training data.
- `"NonLinearityMeasure"`: For the NLM, we need to approximate three line integrals, e.g., via Simpson's 3/8 rule. The general definition of Simpson's 3/8 rule for a univariate function f(x) and integration interval $[a, b]$ corresponds to:

$$\int_a^b f(x) \approx \frac{b-a}{8}\left[f(a) + 3f\left(\frac{2a+b}{3}\right) + 3f\left(\frac{a+2b}{3}\right) + f(b)\right] \tag{3}$$

  We make use of a composite Simpson rule, which divides up the interval $[a, b]$ into n subintervals of equal size and approximates each subinterval with Eq. (3).
- `"Partitioning"`: An abstract class, allowing for various implementations of finding subgroups for cAMEs. For convenience, the wrapper function `came()` can be used. The current implementation supports RP via the **rpart** and **partykit** (CTREE algorithm) packages (classes `"PartitioningRPart"` and `"PartitioningCTREE"`).

  We believe there are two criteria that should guide this process: FME homogeneity within each subgroup and the number of subgroups. A low number of subgroups is generally preferred. In certain applications, we may want to search for a predefined number of subgroups, akin to the search for a predefined number of clusters in clustering problems. Many RP algorithms do not support searching for a number of subgroups, which is what the `"Pruner"` class is intended for.

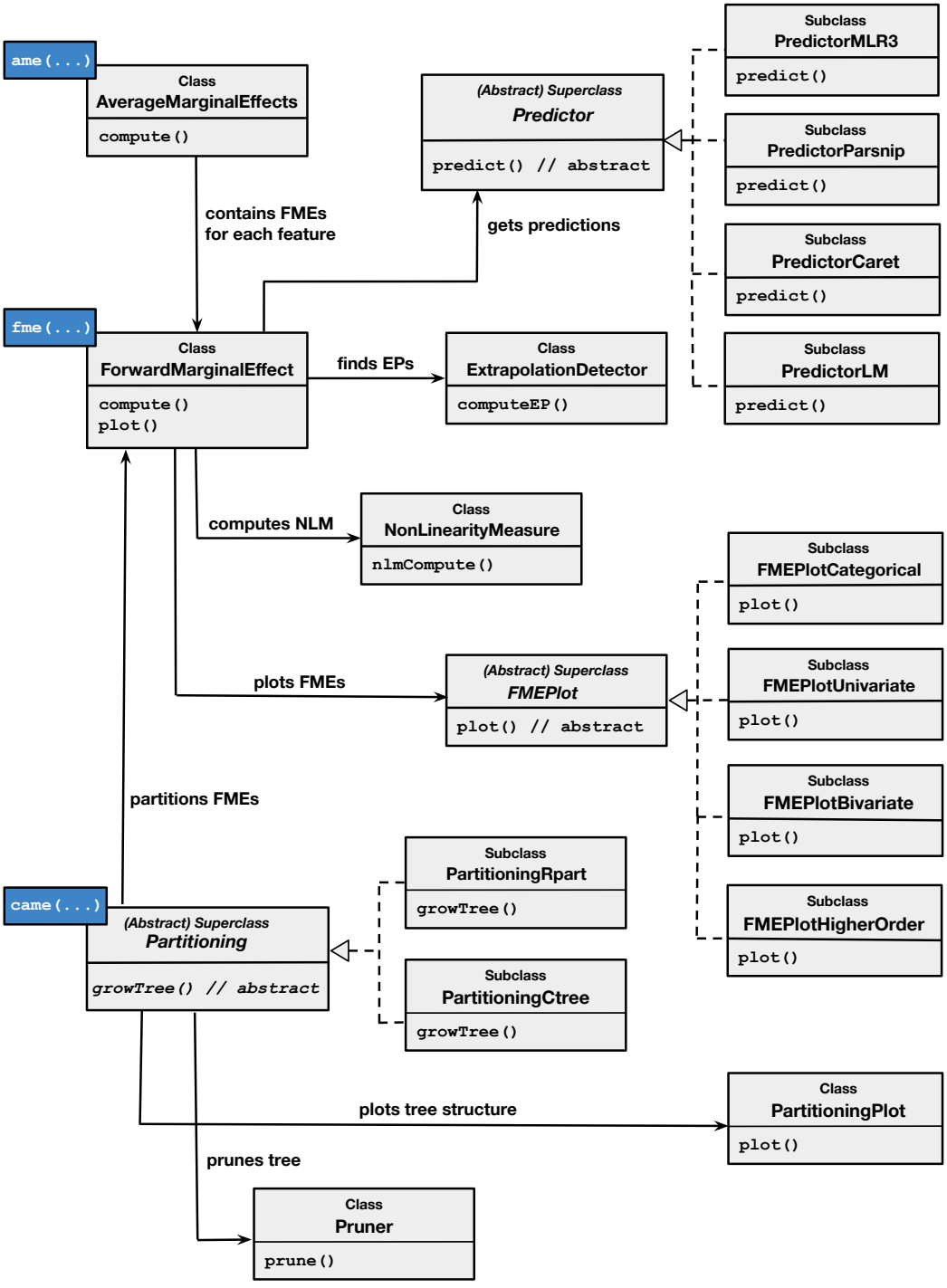

Figure 13: Design overview of the **fmeffects** package, including methods that implement the main functionality of each class. Classes may contain more methods than depicted. Blue boxes indicate wrapper functions to instantiate objects of the respective class.

- `"Pruner"`: To receive a predefined number of subgroups for arbitrary RP algorithms, we follow a two-stage process: grow a large tree by tweaking tree-specific hyperparameters and then prune it back to receive the desired number of subgroups. A `"Partitioning"` subclass is implemented such that it can first grow a large tree, e.g., with a low complexity parameter for **rpart**. Then `"Pruner"` iteratively computes the SD of FMEs for each parent node of the current terminal nodes and removes all terminal nodes of the parent with the lowest SD.
- `"PartitioningPlot"`: Decouples visualizations of the separation of $\mathcal{D}$ into subgroups from specific implementations of the `"Partitioning"` subclass. Here, we make use of a dependency on **partykit** for a tree data structure. This allows visualizations of any partitioning with the same methods. The package **ggparty** (Borkovec and Madin, 2019) creates tree figures that illustrate the partitioning, descriptive statistics for each terminal node, and histograms of FMEs (and optionally NLM values).

8 Conclusion

This paper introduces the R package **fmeffects**, the first software implementation of the theory surrounding FMEs. We showcase the package functionality with an applied use case and discuss design choices and implications for future extensions. FMEs are a versatile model-agnostic interpretation method and give us comprehensible model explanations in the form of: if we change x by an amount h, what is the change in predicted outcome $\widehat{y}$? FMEs equip stakeholders, including those without ML expertise, with the ability to understand feature effects for any model. We therefore hope that this package will work towards a more widespread adoption of FMEs in practice.

Software development is an ongoing process. As the theory surrounding FMEs evolves, so should the **fmeffects** package. As noted by Scholbeck et al. (2024), possible directions for future research include the development of techniques to better quantify extrapolation risk for the selection of step sizes; furthermore, the subgroup search for cAMEs is subject to uncertainties, which may be able to be quantified; and lastly, we may be able to spare computations by searching for representative FMEs, similar to prototype observations that are representative of clusters of observations (Tan et al., 2019). Future performance improvements may also be made via parallel computing, which at this point is only implemented for NLM computations.

References

A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018. URL https://doi.org/10.1109/access.2018.2870052. [p73]

D. W. Apley and J. Zhu. 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, 2020. URL https://doi.org/10.1111/rssb.12377. [p71, 73]

V. Arel-Bundock. *marginaleffects: Predictions, Comparisons, Slopes, Marginal Means, and Hypothesis Tests*, 2023. URL https://CRAN.R-project.org/package=marginaleffects. R package version 0.11.1. [p71]

S. Athey and G. W. Imbens. Machine learning methods that economists should know about. *Annual Review of Economics*, 11(1):685–725, 2019. URL https://doi.org/10.1146/annurev-economics-080217-053433. [p67]

T. Bartus. Estimation of marginal effects using margeff. *The Stata Journal*, 5(3):309 – 329, 2005. [p67, 71]

M. Borkovec and N. Madin. *ggparty: 'ggplot' Visualizations for the 'partykit' Package*, 2019. URL https://CRAN.R-project.org/package=ggparty. R package version 1.0.0. [p85]

A.-L. Boulesteix, M. N. Wright, S. Hoffmann, and I. R. König. Statistical learning approaches in the genetic epidemiology of complex diseases. *Human Genetics*, 139(1):73–84, 2020. URL https://doi.org/10.1007/s00439-019-01996-9. [p67]

L. Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199 – 231, 2001. URL https://doi.org/10.1214/ss/1009213726. [p67, 72]

M. Britton. Vine: Visualizing statistical interactions in black box models. arXiv, 2019. URL https://doi.org/10.48550/arXiv.1904.00561. [p71, 72]

G. Casalicchio, C. Molnar, and B. Bischl. Visualizing the feature importance for black box models. In M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 655–670. Springer International Publishing, Cham, 2019. URL https://doi.org/10.1007/978-3-030-10925-7_40. [p71]

W. Chang. *R6: Encapsulated Classes with Reference Semantics*, 2021. URL https://CRAN.R-project.org/package=R6. R package version 2.5.1. [p73]

I. C. Covert, S. Lundberg, and S.-I. Lee. Understanding global feature contributions with additive importance measures. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. [p71]

P. D. Dueben and P. Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10):3999–4009, 2018. URL https://doi.org/10.5194/gmd-11-3999-2018. [p67]

D. B. Dwyer, P. Falkai, and N. Koutsouleris. Machine learning approaches for clinical psychology and psychiatry. *Annual Review of Clinical Psychology*, 14(1):91–118, 2018. URL https://doi.org/10.1146/annurev-clinpsy-032816-045037. [p67]

H. Fanaee-T. Bike Sharing Dataset. UCI Machine Learning Repository, 2013. URL https://doi.org/10.24432/C5W894. [p73]

J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Ann. Statist.*, 29(5): 1189–1232, 2001. URL https://doi.org/10.1214/aos/1013203451. [p71]

E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1st edition, 1994. [p83]

A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015. URL https://doi.org/10.1080/10618600.2014.907095. [p71]

W. Greene. *Econometric Analysis*. Pearson International, 8th edition, 2019. [p67, 71]

J. Herbinger, B. Bischl, and G. Casalicchio. Repid: Regional effect plots with implicit interaction detection. In G. Camps-Valls, F. J. R. Ruiz, and I. Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 10209–10233. PMLR, 2022. [p71, 72]

G. Hooker. Diagnosing extrapolation: Tree-based density estimation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 569–574, New York, NY, USA, 2004a. Association for Computing Machinery. [p73]

G. Hooker. Discovering additive structure in black box functions. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 575–580, New York, NY, USA, 2004b. ACM. URL http://doi.acm.org/10.1145/1014052.1014122. [p73]

G. Hooker. Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3):709–732, 2007. URL https://doi.org/10.1198/106186007X237892. [p73]

G. Hooker, L. Mentch, and S. Zhou. Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance. *Statistics and Computing*, 31 (6):82, 2021. URL https://doi.org/10.1007/s11222-021-10057-z. [p73]

T. Hothorn and A. Zeileis. partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, 16(118):3905–3909, 2015. [p70]

U. Kamath and J. Liu. Introduction to interpretability and explainability. In *Explainable Artificial Intelligence: An Introduction to Interpretable Machine Learning*, pages 1–26. Springer International Publishing, Cham, 2021. URL https://doi.org/10.1007/978-3-030-83356-5_1. [p67]

M. Kuhn and D. Vaughan. *parsnip: A Common API to Modeling and Analysis Functions*, 2023. URL https://CRAN.R-project.org/package=parsnip. R package version 1.1.1. [p83]

T. J. Leeper. *margins: Marginal effects for model objects*, 2018. URL https://CRAN.R-project.org/package=margins. R package version 0.3.23. [p71]

A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL https://CRAN.R-project.org/doc/Rnews/. [p73]

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. [p71]

D. Lüdecke. ggeffects: Tidy data frames of marginal effects from regression models. *Journal of Open Source Software*, 3(26):772, 2018. URL https://doi.org/10.21105/joss.00772. [p71]

C. J. McCabe, M. A. Halvorson, K. M. King, X. Cao, and D. S. Kim. Interpreting interaction effects in generalized linear models of nonlinear probabilities and counts. *Multivariate Behavioral Research*, 57 (2-3):243–263, 2022. URL https://doi.org/10.1080/00273171.2020.1868966. [p67]

N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), 2021. URL https://doi.org/10.1145/3457607. [p70]

T. D. Mize, L. Doan, and J. S. Long. A general framework for comparing predictions and marginal effects across models. *Sociological Methodology*, 49(1):152–189, 2019. URL https://doi.org/10.1177/0081175019852763. [p71]

C. Molnar. *Interpretable Machine Learning*. 2nd edition, 2022. URL https://christophm.github.io/interpretable-ml-book. [p67, 75]

C. Molnar, B. Bischl, and G. Casalicchio. iml: An R package for interpretable machine learning. *JOSS*, 3(26):786, 2018. URL https://doi.org/10.21105/joss.00786. [p75]

C. Molnar, G. König, J. Herbinger, T. Freiesleben, S. Dandl, C. A. Scholbeck, G. Casalicchio, M. Grosse-Wentrup, and B. Bischl. General pitfalls of model-agnostic interpretation methods for machine learning models. In A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, and W. Samek, editors, *xxAI - Beyond Explainable AI. xxAI 2020. Lecture Notes in Computer Science, vol 13200*, Cham, 2022. Springer. URL https://doi.org/10.1007/978-3-031-04083-2_4. [p73]

C. Molnar, G. König, B. Bischl, and G. Casalicchio. Model-agnostic feature importance and effects with dependent features: A conditional subgroup approach. *Data Mining and Knowledge Discovery*, 38(5): 2903–2941, 2024. URL https://doi.org/10.1007/s10618-022-00901-9. [p71]

S. Mullainathan and J. Spiess. Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017. URL https://doi.org/10.1257/jep.31.2.87. [p67]

E. Onukwugha, J. Bergtold, and R. Jain. A primer on marginal effects—part I: Theory and formulae. *PharmacoEconomics*, 33(1):25–30, 2015. URL https://doi.org/10.1007/s40273-014-0210-6. [p67]

A. Rajkomar, J. Dean, and I. Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019. URL https://doi.org/10.1056/NEJMra1814259. [p67]

M. T. Ribeiro, S. Singh, and C. Guestrin. "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*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. URL https://doi.org/10.1145/2939672.2939778. [p71]

C. A. Scholbeck, C. Molnar, C. Heumann, B. Bischl, and G. Casalicchio. Sampling, intervention, prediction, aggregation: A generalized framework for model-agnostic interpretations. In P. Cellier and K. Driessens, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 1167 of *Communications in Computer and Information Science*, pages 205–216. Springer International Publishing, Cham, 2020. URL https://doi.org/10.1007/978-3-030-43823-4_18. [p67, 71]

C. A. Scholbeck, J. Moosbauer, G. Casalicchio, H. Gupta, B. Bischl, and C. Heumann. Position paper: Bridging the gap between machine learning and sensitivity analysis. arXiv, 2023. URL https://doi.org/10.48550/arXiv.2312.13234. [p71]

C. A. Scholbeck, G. Casalicchio, C. Molnar, B. Bischl, and C. Heumann. Marginal effects for non-linear prediction functions. *Data Mining and Knowledge Discovery*, 38(5):2997–3042, 2024. URL https://doi.org/10.1007/s10618-023-00993-x. [p67, 68, 69, 70, 71, 73, 85]

StataCorp. *Stata: Release 18*. College Station, TX: StataCorp LLC., 2023. [p71]

E. Štrumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(1):1–18, 2010. [p71]

P.-N. Tan, A. Karpatne, M. Steinbach, and V. Kumar. *Introduction to Data Mining: Global Edition*. Pearson, 2019. [p85]

T. Therneau and B. Atkinson. *rpart: Recursive Partitioning and Regression Trees*, 2019. URL https://CRAN.R-project.org/package=rpart. R package version 4.1-15. [p70]

S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law and Technology*, 31(2):841–887, 2018. [p71]

R. Williams. Using the margins command to estimate and interpret adjusted predictions and marginal effects. *Stata Journal*, 12(2):308–331(24), 2012. [p67, 68, 71]

Holger Löwe
Ludwig-Maximilians-Universität in Munich
Germany
hbj.loewe@gmail.com

Christian A. Scholbeck
Ludwig-Maximilians-Universität in Munich
Munich Center for Machine Learning (MCML)
Germany
https://orcid.org/0000-0001-6607-4895
christian.scholbeck@stat.uni-muenchen.de

Christian Heumann
Ludwig-Maximilians-Universität in Munich
Germany
christian.heumann@stat.uni-muenchen.de

Bernd Bischl
Ludwig-Maximilians-Universität in Munich
Munich Center for Machine Learning (MCML)
Germany
bernd.bischl@stat.uni-muenchen.de

Giuseppe Casalicchio
Ludwig-Maximilians-Universität in Munich
Munich Center for Machine Learning (MCML)
Germany
giuseppe.casalicchio@stat.uni-muenchen.de