NORSK REGNESENTRAL
NORWEGIAN COMPUTING CENTER

shapr – Conditional Shapley Value Explanation in R and Python

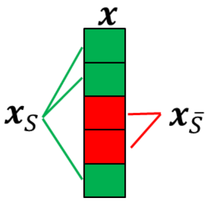
Workshop: Methods for Explainable Machine Learning in Health Care

Code examples available at
https://github.com/NorskRegnesentral/shapr/tree/master/inst/demo

Martin Jullum, Senior Research Scientist, Norwegian Computing Center

Feb 4th 2026

Background



$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!\,(|M| - |S| - 1)}{|M|!} \big(v(S \cup \{j\}) - v(S)\big)$$

$$\begin{aligned} v(S) &= E[f(\boldsymbol{x}) | \boldsymbol{x}_S = \boldsymbol{x}_S^*] \\ &= \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*) \mathrm{d}\boldsymbol{x}_{\bar{S}} \end{aligned}$$

- 2017/2018: Developed interest in explainability
- 2018: Identified dependence issue in *Lundberg & Lee (2017)* $\longrightarrow$ $v(S) \approx \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}) \mathrm{d}\boldsymbol{x}_{\bar{S}}$
- 2019: First preprint of *Aas, Jullum & Løland (2021)*
- 2020: JOSS paper for *shapr 0.2.3, Sellereite & Jullum (2020)*
- 2020-2024: Multiple methodological papers for estimating $v(S)$ ++
- 2024-2025: Complete overhaul of *shapr* + Python wrapper (*shaprpy*)
- 2025: Preprint of software paper for *shapr* + *shaprpy*

Core idea of *shapr*: Contrast *shap* in Python – doing dependence aware estimation of $v(S)$

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *NeurIPS*.

Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*

Sellereite, N., & Jullum, M. (2020). shapr: An R-package for explaining machine learning models with dependence-aware Shapley values. *Journal of Open Source Software*

KernelSHAP WLS problem

$$\underset{\phi\in\mathbb{R}^{M+1}}{\arg\min} \sum_{\mathcal{S}\subseteq\mathcal{M}} k(M,|\mathcal{S}|)\Big(\phi_0+\sum_{j\in\mathcal{S}}\phi_j - v(\mathcal{S})\Big)^2$$

Estimation procedure in *shapr*

- Use KernelSHAP to approximate the Shapley value formula

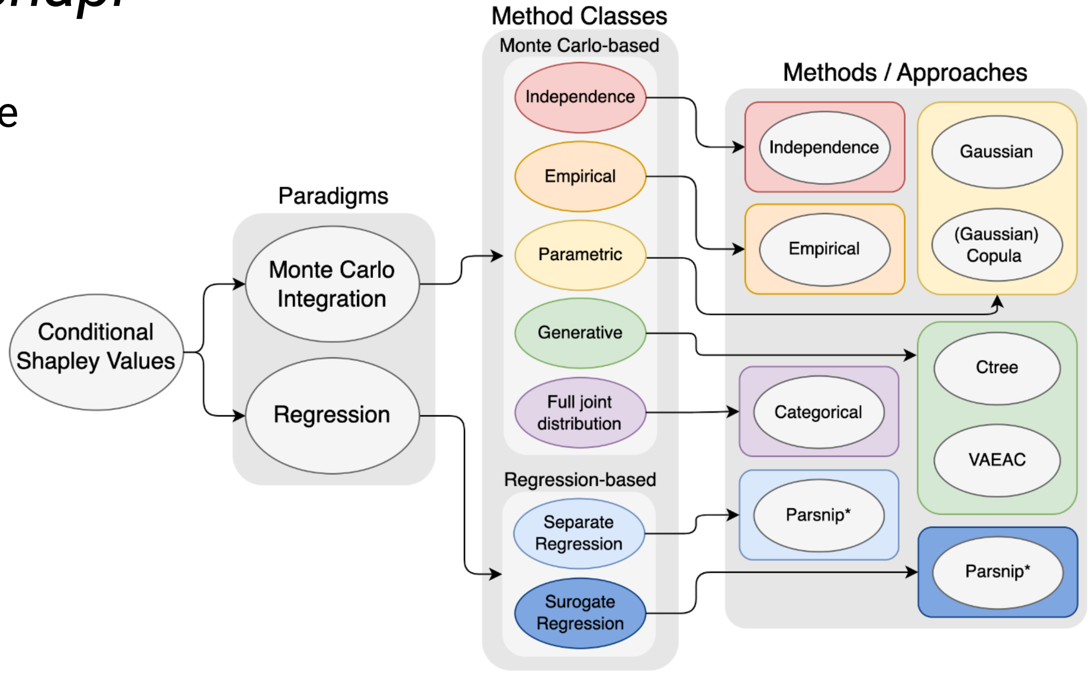
A wide range of approaches for estimating $v(\mathcal{S})$

- Monte Carlo integration

$$v(\mathcal{S}) = \mathbb{E}_{\boldsymbol{x}_{\bar{\mathcal{S}}}}\left[f(\boldsymbol{x}_{\bar{\mathcal{S}}}, \boldsymbol{x}_{\mathcal{S}})|\boldsymbol{x}_{\mathcal{S}} = \boldsymbol{x}_{\mathcal{S}}^*\right] \approx \frac{1}{K}\sum_{k=1}^{K} f(\boldsymbol{x}_{\bar{\mathcal{S}}}^{(k)}, \boldsymbol{x}_{\mathcal{S}}^*)$$

- Regression $f(\boldsymbol{x}_{\bar{\mathcal{S}}}, \boldsymbol{x}_{\mathcal{S}}) \sim \boldsymbol{x}_{\mathcal{S}}$

$$v(\mathcal{S}) = \underset{c}{\arg\min}\, \mathbb{E}_{\boldsymbol{x}_{\bar{\mathcal{S}}}}\left[(f(\boldsymbol{x}_{\bar{\mathcal{S}}}, \boldsymbol{x}_{\mathcal{S}}) - c)^2|\boldsymbol{x}_{\mathcal{S}} = \boldsymbol{x}_{\mathcal{S}}^*\right]$$



Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*

Aas, K., Nagler, T., Jullum, M., & Løland, A. (2021). Explaining predictive models using Shapley values and non-parametric vine copulas. *Dependence Modeling*

Redelmeier, A., Jullum, M., & Aas, K. (2020). Explaining predictive models with mixed features using Shapley values and conditional inference trees. *Int. Cross-Domain Conf for ML & KE.*

Olsen, L. H., Glad, I. K., Jullum, M., & Aas, K. (2022). Using Shapley values and variational autoencoders to explain predictive models with dependent mixed features. *JMLR*

Olsen, L. H. B., Glad, I. K., Jullum, M., & Aas, K. (2024). A comparative study of methods for estimating model-agnostic Shapley value explanations. *DM & KD*

Package capabilities

- Native support for *lm, glm, xgboost, ranger, mgcv-gam, parsnip/workflows, …*
- Custom model support
- Bootstrapping coalition sampling uncertainty
- Convergence detection
- Iterative estimation and continued estimation
- MSEv evaluation of estimation approaches
- Feature group explanation
- Asymmetric and Causal Shapley values

- Improved KernelSHAP efficiency
- Parallelized batch computations (*future*)
- Progress reports and information while running (*progressr* + *cli*)
- Utilize C++ (*Rcpp/RcppArmadillo*) for computationally demanding code segments
- Visualization of explanations (*ggplot2*)
- Explain time series forecasts (multiple horizons)
- Approaches can be combined – one approach per size of |S|

Covert, I., & Lee, S. I. (2021). Improving kernelshap: Practical shapley value estimation using linear regression. AISTATS

Frye, C., et al.(2020). Shapley explainability on the data manifold. *ICLR*

Jullum, M., Redelmeier, A., & Aas, K. (2021). Efficient and simple prediction explanations with groupShapley: A practical perspective. In *Proc of the 2nd Italian Workshop on XAI*.

Heskes, T. et al. (2020). Causal shapley values: Exploiting causal knowledge to explain individual predictions of complex models. NeurIPS

Frye, C., Rowat, C., & Feige, I. (2020). Asymmetric shapley values: incorporating causal knowledge into model-agnostic explainability. NeurIPS.

Olsen, L. H. B., & Jullum, M. (2025). Improving the Weighting Strategy in KernelSHAP. In *World Conference on Explainable Artificial Intelligence*.

shaprpy – the Python wrapper

- Lightweight Python wrapper around *shapr* for explaining models fitted in Python
- Built using the *rpy2* library
- Visualization through the *shap* library
- All core capabilities of the *shapr* package
- Exceptions
    - Explaining time series forecasts
    - Parallelization
- Available on PyPI

Getting started

- Pkgdown site norskregnesentral.github.io/shapr
  - Lots of examples in 4 comprehensive vignettes
  - Full installation instruction
- *shapr* – install from CRAN: *install.packages('shapr')*
- *shaprpy* – install from PyPI: *pip install shaprpy*
  - Requires access to R with shapr installed
- Software paper (arxiv preprint)
- Code and issues: github.com/NorskRegnesentral/shapr

Jullum, M., Olsen, L. H. B., Lachmann, J., & Redelmeier, A. (2025). shapr: Explaining Machine Learning Models with Conditional Shapley Values in R and Python. *arXiv preprint arXiv:2504.01842*.

shapr 1.0.8 Vignettes News Manual Python Search for

shapr

NORSK REGNESENTRAL NORWEGIAN COMPUTING CENTER

See the pkgdown site at norskregnesentral.github.io/shapr/ for a complete introduction with examples and documentation of the package.

For an overview of the methodology and capabilities of the package (per `shapr` v1.0.8), see the software paper Jullum et al. (2025), available as a preprint here.

NEWS

With `shapr` version 1.0.0 (GitHub only, Nov 2024) and version 1.0.1 (CRAN, Jan 2025), the package underwent a major update, providing a full restructuring of the code base, and a full suite of new functionality, including:

- A long list of approaches for estimating the contribution/value function $v(S)$, including Variational Autoencoders and regression-based methods
- Iterative Shapley value estimation with convergence detection
- Parallelized computations with progress updates
- Reweighted Kernel SHAP for faster convergence
- New function `explain_forecast()` for explaining forecasts
- Asymmetric and causal Shapley values
- Several other methodological, computational and user-experience improvements
- Python wrapper `shaprpy` making the core functionality of `shapr` available in Python

See the NEWS for a complete list.

Coming from shapr < 1.0.0?

`shapr` version >= 1.0.0 comes with a number of breaking changes. Most notably, we moved from using two functions (`shapr()` and `explain()`) to one function (`explain()`). In addition, custom models are now explained by passing the prediction function directly to `explain()`. Several input arguments were renamed, and a few functions for edge cases were removed to simplify the code base.

Click here to view a version of this README with the old syntax (v0.2.2).

Python wrapper

We provide a Python wrapper (`shaprpy`) which allows explaining Python models with the methodology implemented in `shapr`, directly from Python. The wrapper calls R internally and therefore requires an installation of R. See here for installation instructions and examples.

The package

The `shapr` R package implements an enhanced version of the Kernel SHAP method for approximating Shapley values, with a strong focus on conditional Shapley values. The core idea is to remain completely model-agnostic while offering a variety of methods for estimating contribution functions, enabling accurate computation of conditional Shapley values across different feature types, dependencies, and distributions. The package also includes evaluation metrics to compare various approaches. With features like parallelized computations, convergence detection, progress updates, and extensive plotting options, shapr is a highly efficient and user-friendly tool, delivering precise estimates of conditional Shapley values, which are critical for understanding how features truly contribute to predictions.

A basic example is provided below. Otherwise, we refer to the pkgdown website and the vignettes there for details and further examples.

Installation

`shapr` is available on CRAN and can be installed in R as:

Links
View on CRAN
Browse source code
Report a bug

License
Full license
MIT + file LICENSE

Community
Contributing guide
Code of conduct

Citation
Citing shapr

Developers
Martin Jullum
Maintainer, author
Lars Henry Berge Olsen
Author
Annabelle Redelmeier
Author
Jon Lachmann
Author
Nikolai Sellereite
Author
Norsk Regnesentral
Copyright holder, funder
More about authors...

Dev status
CRAN 1.0.8 – 2026-01-20
downloads 105K
R-CMD-check passing
lifecycle stable
License MIT
JOSS (v0.2.3) 10.21105/joss.02027
arXiv (v1.0.8) 2504.01842

DEMO TIME!

Code examples available at
https://github.com/NorskRegnesentral/shapr/tree/master/inst/demo