Random Planted Forest
A Directly Interpretable Tree Ensemble

Meyer, J. T.[5] Burk, L.[1,2,3,4] Hiabu, M.[6] Mammen, E.[5]

[1]Leibniz Institute for Prevention Research and Epidemiology – BIPS
[2]LMU Munich [3]University of Bremen
[4]Munich Center for Machine Learning (MCML)
[5]Heidelberg University [6]University of Copenhagen

DAGStat 2025 — March 27th, 2025

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
  - Fast & flexible

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
  - Fast & flexible
  - Interpretable? → It depends

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
  - Fast & flexible
  - Interpretable? → It depends
- Desirable properties:

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
    - Fast & flexible
    - Interpretable? → It depends
- Desirable properties:
    - Meaningful feature importance

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
  - Fast & flexible
  - Interpretable? → It depends
- Desirable properties:
  - Meaningful feature importance
  - Quantification of main- and interaction effects

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
  - Fast & flexible
  - Interpretable? → It depends
- Desirable properties:
  - Meaningful feature importance
  - Quantification of main- and interaction effects
- Additive models (LM, GAM, ...) can provide both
  → Need to manually specify interactions in model fit

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
    - Fast & flexible
    - Interpretable? → It depends
- Desirable properties:
    - Meaningful feature importance
    - Quantification of main- and interaction effects
- Additive models (LM, GAM, ...) can provide both
  → Need to manually specify interactions in model fit

Motivation

- Goal: Fast, flexible, and **interpretable** predictive models
- Tree-based methods like Random Forest (RF):
    - Fast & flexible
    - Interpretable? → It depends
- Desirable properties:
    - Meaningful feature importance
    - Quantification of main- and interaction effects
- Additive models (LM, GAM, ...) can provide both
  → Need to manually specify interactions in model fit

→ **Random Planted Forest** (RPF): Additive Random Forest

Functional ANOVA Expansion

- Setting: Regression with target $Y_i \in \mathbb{R}$, features $X_i \in \mathbb{R}^p$, instance $\mathbf{x}_i$

Functional ANOVA Expansion

- Setting: Regression with target $Y_i \in \mathbb{R}$, features $X_i \in \mathbb{R}^p$, instance $\mathbf{x}_i$
- Expand prediction $\mathbb{E}(Y_i | X = \mathbf{x}_i) = \hat{m}(\mathbf{x}_i)$ into

Functional ANOVA Expansion

- Setting: Regression with target $Y_i \in \mathbb{R}$, features $X_i \in \mathbb{R}^p$, instance $\mathbf{x}_i$
- Expand prediction $\mathbb{E}(Y_i|X = \mathbf{x}_i) = \hat{m}(\mathbf{x}_i)$ into
  - $\hat{m}_0$: Average prediction ("intercept")

Functional ANOVA Expansion

- Setting: Regression with target $Y_i \in \mathbb{R}$, features $X_i \in \mathbb{R}^p$, instance $\mathbf{x}_i$
- Expand prediction $\mathbb{E}(Y_i | X = \mathbf{x}_i) = \hat{m}(\mathbf{x}_i)$ into
  - $\hat{m}_0$: Average prediction ("intercept")
  - Terms $\hat{m}_S$ with feature set $S \subseteq \{1, \dots, p\}$

Functional ANOVA Expansion

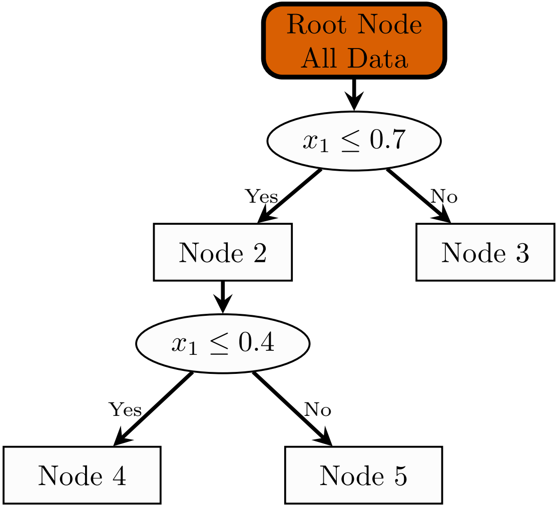
- Setting: Regression with target $Y_i \in \mathbb{R}$, features $X_i \in \mathbb{R}^p$, instance $\mathbf{x}_i$
- Expand prediction $\mathbb{E}(Y_i | X = \mathbf{x}_i) = \hat{m}(\mathbf{x}_i)$ into
  - $\hat{m}_0$: Average prediction ("intercept")
  - Terms $\hat{m}_S$ with feature set $S \subseteq \{1, \ldots, p\}$

Functional ANOVA Expansion

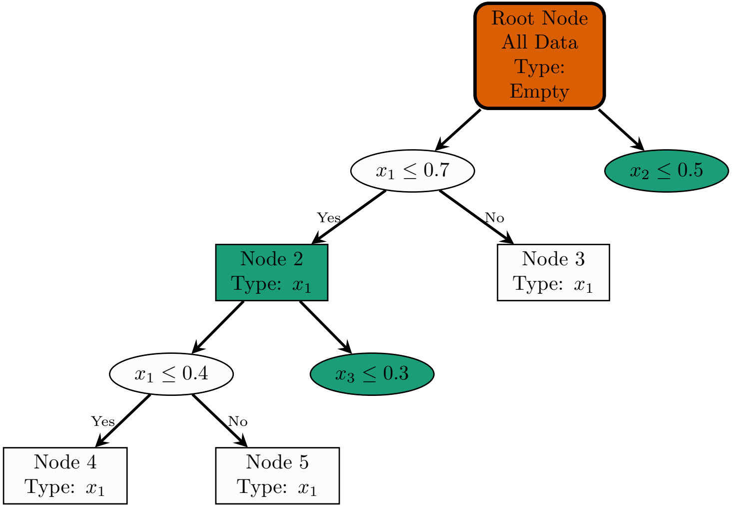
- Setting: Regression with target $Y_i \in \mathbb{R}$, features $X_i \in \mathbb{R}^p$, instance $\mathbf{x}_i$
- Expand prediction $\mathbb{E}(Y_i|X = \mathbf{x}_i) = \hat{m}(\mathbf{x}_i)$ into
  - $\hat{m}_0$: Average prediction ("intercept")
  - Terms $\hat{m}_S$ with feature set $S \subseteq \{1, \ldots, p\}$

$$
\begin{aligned}
\hat{m}(\mathbf{x}_i) = & \hat{m}_0 + \\
& \underbrace{\hat{m}_1(x_1) + \hat{m}_2(x_2) + \hat{m}_3(x_3)}_{\text{Main effect terms}} + \\
& \underbrace{\hat{m}_{1,2}(x_1, x_2) + \hat{m}_{1,3}(x_1, x_3) + \hat{m}_{2,3}(x_2, x_3)}_{\text{2nd order interactions}} + \\
& \underbrace{\hat{m}_{1,2,3}(x_1, x_2, x_3)}_{\text{3rd order interaction}}
\end{aligned}
$$

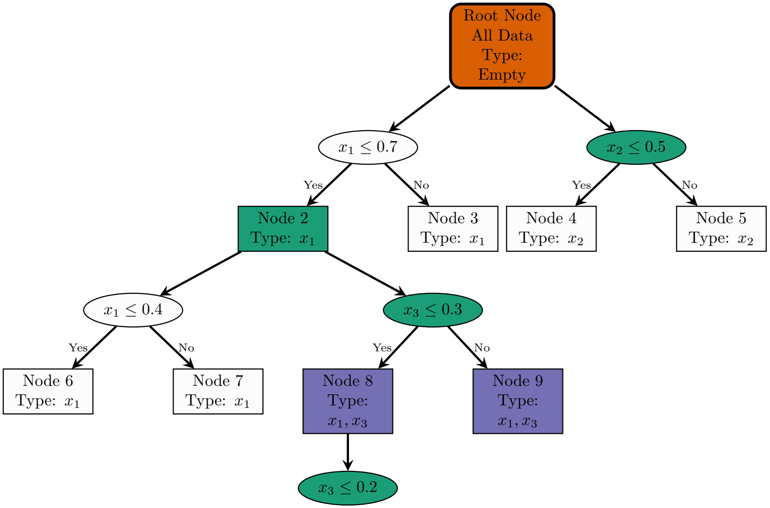
Trees in Random Forest



Planted Trees (I)



Planted Trees (II)



Key features of Random Planted Forests

- Ensemble of trees (like RF)

Key features of Random Planted Forests

- Ensemble of trees (like RF)
- Splits nodes multiple times (→ non-binary trees!)

Key features of Random Planted Forests

- Ensemble of trees (like RF)
- Splits nodes multiple times (→ non-binary trees!)
- Nodes keep track of features involved in construction

Key features of Random Planted Forests

- Ensemble of trees (like RF)
- Splits nodes multiple times ($\rightarrow$ non-binary trees!)
- Nodes keep track of features involved in construction
- Degree of interaction can be constrained

Key features of Random Planted Forests

- Ensemble of trees (like RF)
- Splits nodes multiple times (→ non-binary trees!)
- Nodes keep track of features involved in construction
- Degree of interaction can be constrained
- Tree stops after adjustable number of splits

Key features of Random Planted Forests

- Ensemble of trees (like RF)
- Splits nodes multiple times (→ non-binary trees!)
- Nodes keep track of features involved in construction
- Degree of interaction can be constrained
- Tree stops after adjustable number of splits
- Prediction built up incrementally, guided by residuals (cf. Gradient Boosting)

Application Example

- `Bikeshare` regression dataset [1]

[1] from UCI ML repository

Application Example

- `Bikeshare` regression dataset [1]
- Target `bikers`: Number of bikers on a given day in 2011/2012

[1] from UCI ML repository

Application Example

- `Bikeshare` regression dataset [1]
- Target **`bikers`**: Number of bikers on a given day in 2011/2012
- Focus on 3 features for example

[1] from UCI ML repository

Application Example

- `Bikeshare` regression dataset [1]
- Target `bikers`: Number of bikers on a given day in 2011/2012
- Focus on 3 features for example
  - `hour` of day $\in \{0, 1, \dots, 23\}$

[1] from UCI ML repository

Application Example

- `Bikeshare` regression dataset [1]
- Target `bikers`: Number of bikers on a given day in 2011/2012
- Focus on 3 features for example
  - `hour` of day $\in \{0, 1, \dots, 23\}$
  - `temp` normalized temperature $\in [0, 1]$

[1] from UCI ML repository

Application Example

- `Bikeshare` regression dataset [1]
- Target `bikers`: Number of bikers on a given day in 2011/2012
- Focus on 3 features for example
  - `hour` of day $\in \{0, 1, \dots, 23\}$
  - `temp` normalized temperature $\in [0, 1]$
  - `workingday` binary → {`workingday`, `no workingday`}

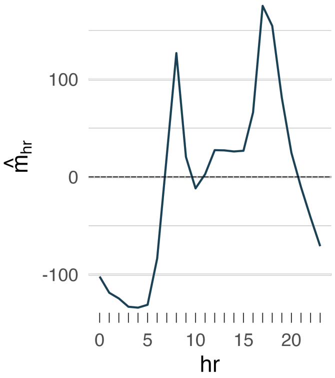
[1] from UCI ML repository

Application Example

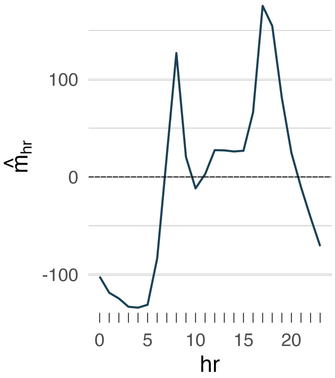
- `Bikeshare` regression dataset [1]
- Target `bikers`: Number of bikers on a given day in 2011/2012
- Focus on 3 features for example
  - `hour` of day $\in \{0, 1, \ldots, 23\}$
  - `temp` normalized temperature $\in [0, 1]$
  - `workingday` binary → {`workingday`, `no workingday`}
- Average prediction: $\hat{m}_0 \approx 144$

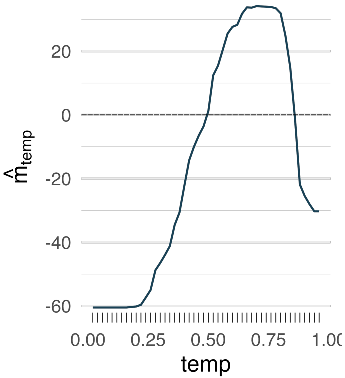
[1] from UCI ML repository

Main Effects

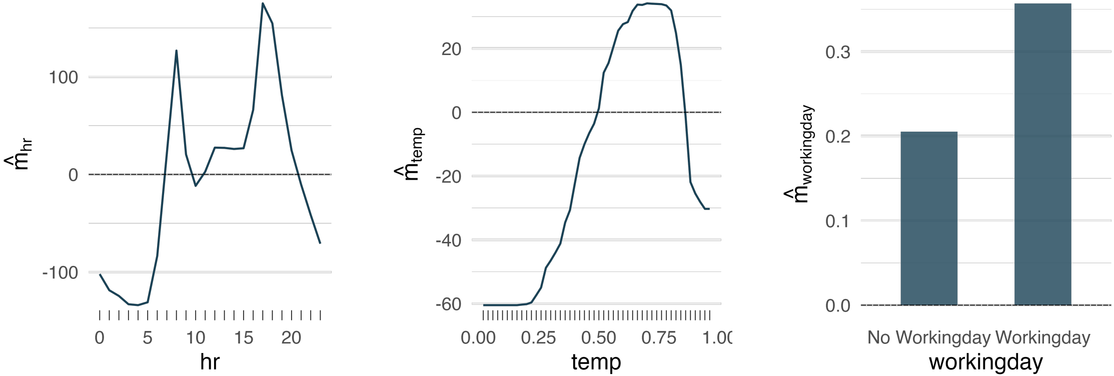


Main Effects

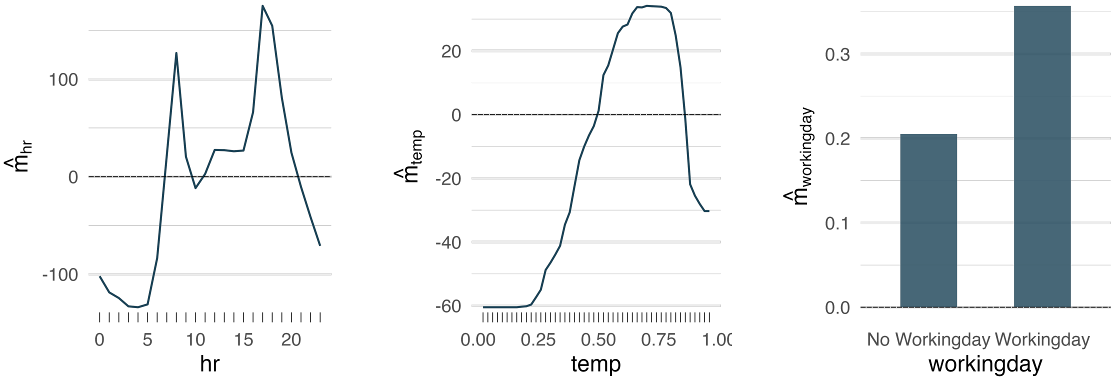




Main Effects

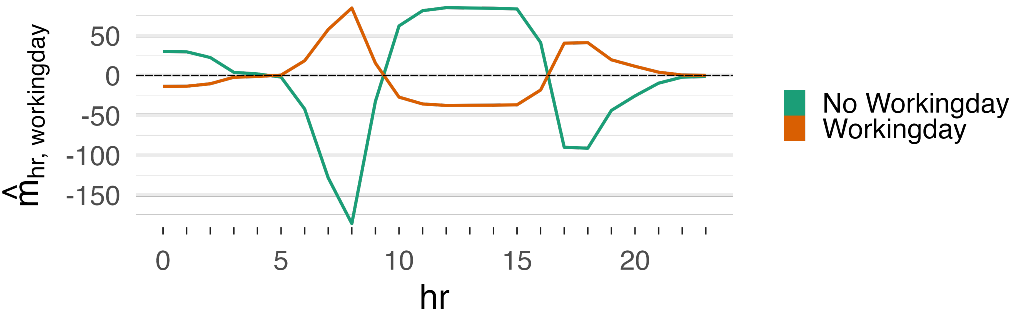


Main Effects



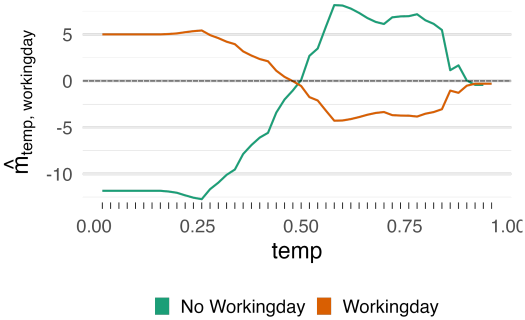
$$\hat{m} = \hat{m}_0 + \hat{m}_{\mathtt{hr}}(\mathtt{hr}) + \hat{m}_{\mathtt{temp}}(\mathtt{temp}) + \hat{m}_{\mathtt{workingday}}(\mathtt{workingday}) + \dots$$

Hour × Working Day: "Rush Hour Effect"



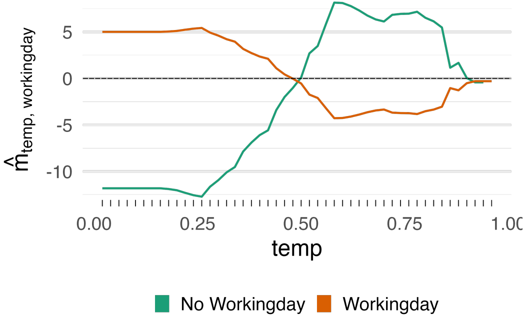
$$\ldots + \hat{m}_{\texttt{hr,workingday}}(\texttt{hr}, \texttt{workingday}) + \ldots$$

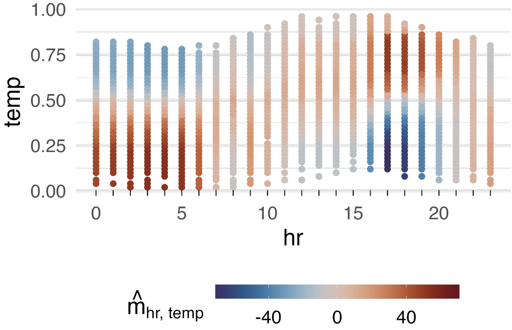
More 2nd Order Interactions



$$+\hat{m}_{\texttt{temp,workingday}}(\texttt{temp}, \texttt{workingday})$$

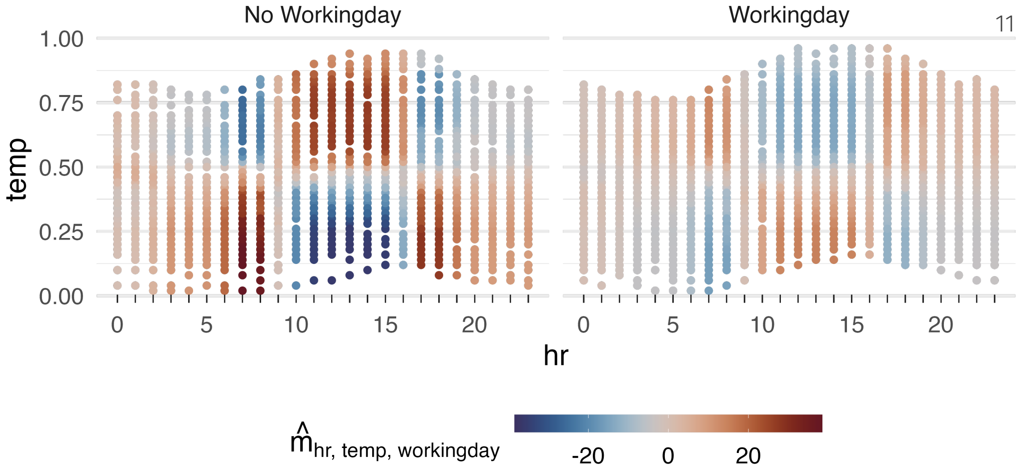
More 2nd Order Interactions





$$+\hat{m}_{\texttt{temp,workingday}}(\texttt{temp}, \texttt{workingday}) \qquad +\hat{m}_{\texttt{hr,temp}}(\texttt{hr}, \texttt{temp}) + \ldots$$

3rd Order Interaction



Feature Importance in RPF

- Average of absolute values of term $\hat{m}_S$ of interest

$$\mathrm{FI}_S = \frac{1}{n}\sum_{i=1}^{n} |\hat{m}_S(\mathbf{x}_i)|$$

Feature Importance in RPF

- Average of absolute values of term $\hat{m}_S$ of interest

$$\mathrm{FI}_S = \frac{1}{n} \sum_{i=1}^{n} |\hat{m}_S(\mathbf{x}_i)|$$

- Unlike RF Feature importance:

Feature Importance in RPF

- Average of absolute values of term $\hat{m}_S$ of interest

$$\mathrm{FI}_S = \frac{1}{n}\sum_{i=1}^{n} |\hat{m}_S(\mathbf{x}_i)|$$

- Unlike RF Feature importance:
  - Scores also per interaction term

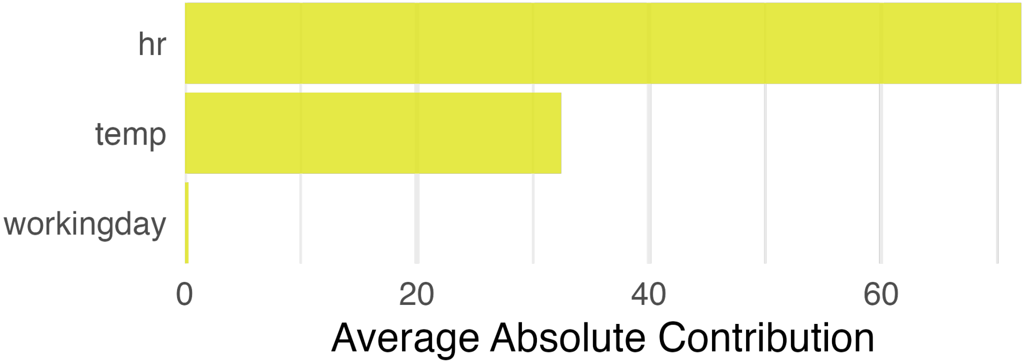
Feature Importance in RPF

- Average of absolute values of term $\hat{m}_S$ of interest

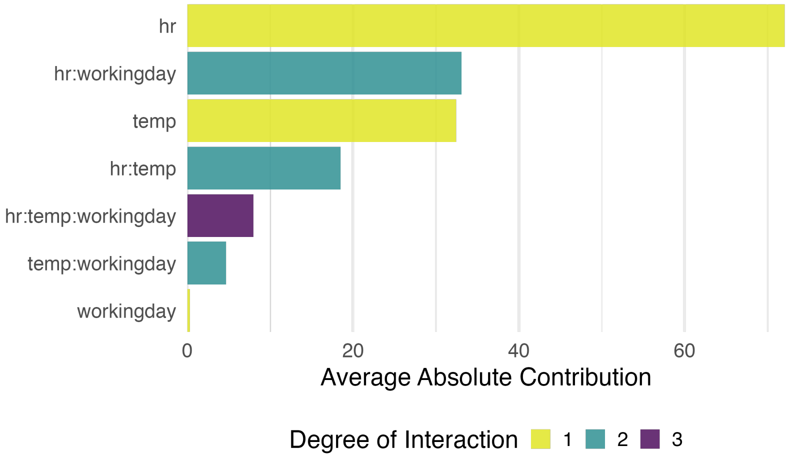
$$\mathrm{FI}_S = \frac{1}{n}\sum_{i=1}^{n} |\hat{m}_S(\mathbf{x}_i)|$$

- Unlike RF Feature importance:
  - Scores also per interaction term
  - Importance scores on same scale as prediction

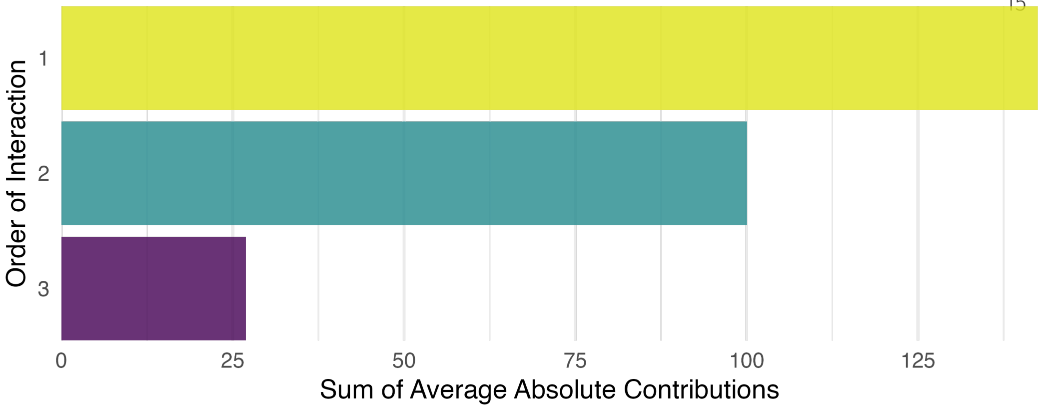
Feature Importance per Main Term



Feature Importance for All Terms



Feature Importance by Order of Interaction



No Free Lunch

Gains in interpretibility → sacrifices in predictive performance?

No Free Lunch

Gains in interpretibility → sacrifices in predictive performance?

- Benchmark on 28 datasets [2] comparing RPF with XGBoost & RF, incl. tuning

[2] OpenML-CTR23 regression benchmark suite: Fischer et al. (2023)

No Free Lunch

Gains in interpretibility → sacrifices in predictive performance?

- Benchmark on 28 datasets [2] comparing RPF with XGBoost & RF, incl. tuning
- Additionally comparing XGBoost & RPF with constrained order of interaction to 2

[2] OpenML-CTR23 regression benchmark suite: Fischer et al. (2023)

No Free Lunch

Gains in interpretibility → sacrifices in predictive performance?

- Benchmark on 28 datasets [2] comparing RPF with XGBoost & RF, incl. tuning
- Additionally comparing XGBoost & RPF with constrained order of interaction to 2

[2] OpenML-CTR23 regression benchmark suite: Fischer et al. (2023)

No Free Lunch

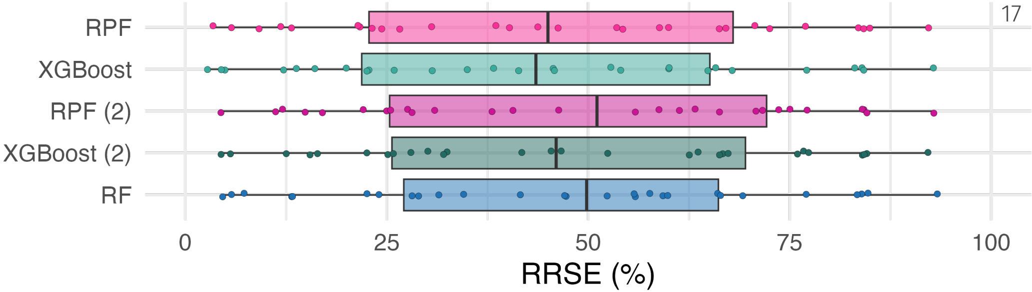
Gains in interpretibility → sacrifices in predictive performance?

- Benchmark on 28 datasets [2] comparing RPF with XGBoost & RF, incl. tuning
- Additionally comparing XGBoost & RPF with constrained order of interaction to 2

→ Generally: RPF never best, rarely bad, usually close

[2] OpenML-CTR23 regression benchmark suite: Fischer et al. (2023)

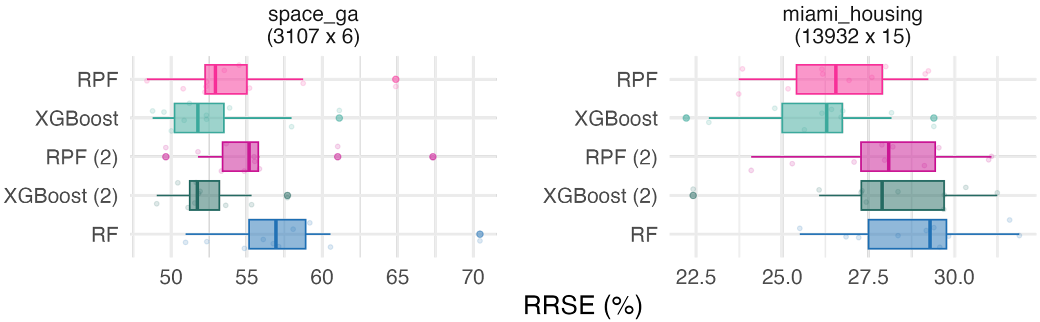
Benchmark Results (Aggregated)



$$\mathrm{RRSE} := \sqrt{\mathrm{SSE}(Y, \hat{Y}) \;/\; \mathrm{SSE}(Y, \bar{Y})}$$

→ Featureless model scores 1, perfect score 0

Benchmark Results (Selected Tasks)



Summary

Random Planted Forests = Additive Random Forests

- ($\uparrow$) Nicely interpretable feature importance on same scale as target

Summary

Random Planted Forests = Additive Random Forests

- ($\uparrow$) Nicely interpretable feature importance on same scale as target
- ($\uparrow$) Quantifies main effects and interaction terms

Summary

Random Planted Forests = Additive Random Forests

- (↑) Nicely interpretable feature importance on same scale as target
- (↑) Quantifies main effects and interaction terms
- (↑) R package available [3]

[3] github.com/PlantedML/randomPlantedForest

Summary

Random Planted Forests = Additive Random Forests

- (↑) Nicely interpretable feature importance on same scale as target
- (↑) Quantifies main effects and interaction terms
- (↑) R package available [3]
- (→) Competetive predictive performance (mostly)

[3] github.com/PlantedML/randomPlantedForest

Summary

Random Planted Forests = Additive Random Forests

- (↑) Nicely interpretable feature importance on same scale as target
- (↑) Quantifies main effects and interaction terms
- (↑) R package available [3]
- (→) Competetive predictive performance (mostly)
- (↓) Computationally heavy for large data (Optimization WIP!)

[3] github.com/PlantedML/randomPlantedForest

Thank you for your attention!

www.leibniz-bips.de/en

Contact
Lukas Burk
Leibniz Institute for Prevention Research
and Epidemiology – BIPS
Achterstraße 30
D-28359 Bremen
burk@leibniz-bips.de

References I

Fischer, Sebastian Felix et al. (2023). “OpenML-CTR23 – A Curated Tabular Regression Benchmarking Suite”. In: AutoML Conference 2023 (Workshop).