Supervised learning with missing values and imputation

Marine Le Morvan – Soda, INRIA

Supervised learning and missing values

Question?

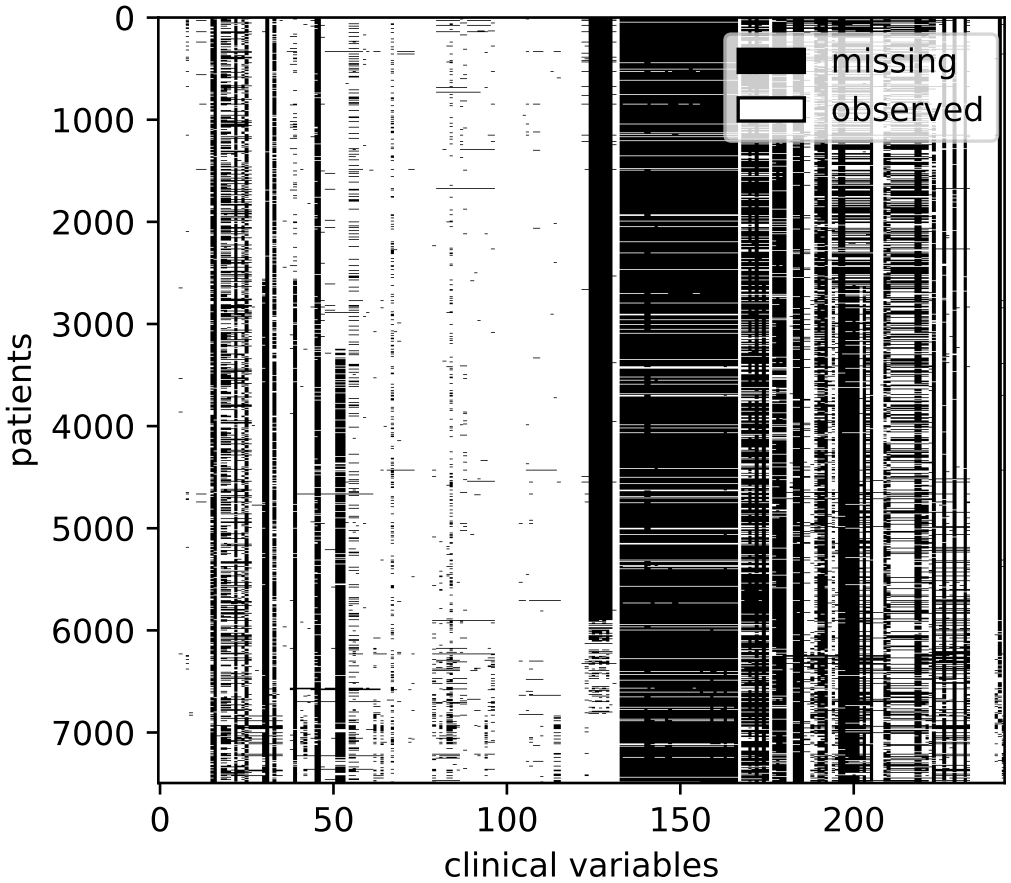
Is this patient at risk of hemorrhagic shock?

Task:

Classification on incomplete data:

$$\begin{pmatrix} 1.2 & \texttt{na} & 0.2 \\ -0.3 & \texttt{na} & \texttt{na} \\ \texttt{na} & 0.4 & \texttt{na} \\ -2.8 & 1.9 & 1.6 \\ \texttt{na} & 1.7 & -2.4 \end{pmatrix} \qquad \begin{pmatrix} 1.7 \\ -0.2 \\ 1.6 \\ -2.2 \\ 0.8 \end{pmatrix}$$

Incomplete X Complete Y



Problem:

How to deal with NAs at **train** and **inference** time?

traumabase.eu

Challenges of supervised learning with missing values

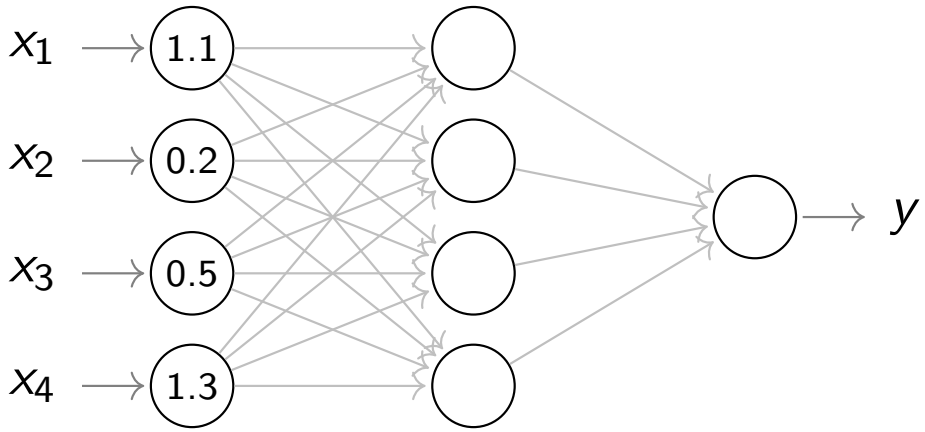
- Supervised learning with missing values:
  - Scarcer literature (vs Inference, Imputation)

Challenges of supervised learning with missing values

- Supervised learning with missing values:
  - Scarcer literature (vs Inference, Imputation)
  - Requires handling arbitrary subsets of variables.

Samples

$$\begin{bmatrix} 1.1 & 0.2 & 0.5 & 1.3 \\ & & & \end{bmatrix}$$

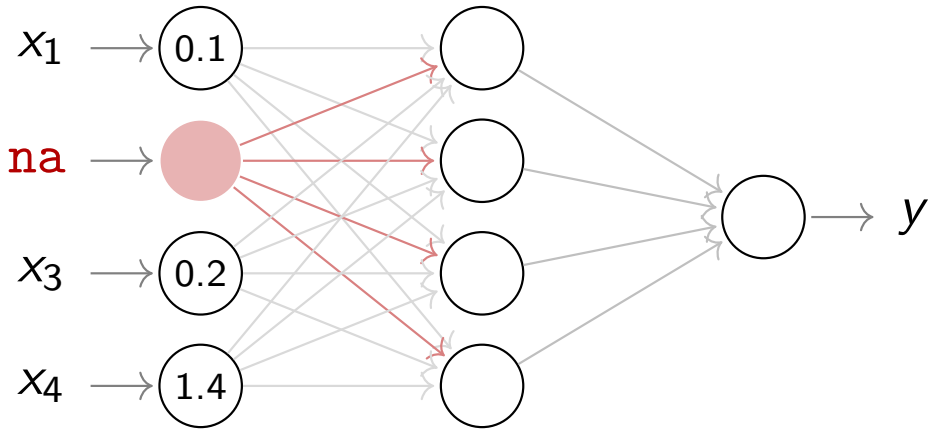


Challenges of supervised learning with missing values

- Supervised learning with missing values:
  - Scarcer literature (vs Inference, Imputation)
  - Requires handling arbitrary subsets of variables.

Samples

$$\begin{bmatrix} 1.1 & 0.2 & 0.5 & 1.3 \\ 0.1 & \texttt{na} & 0.2 & 1.4 \end{bmatrix}$$

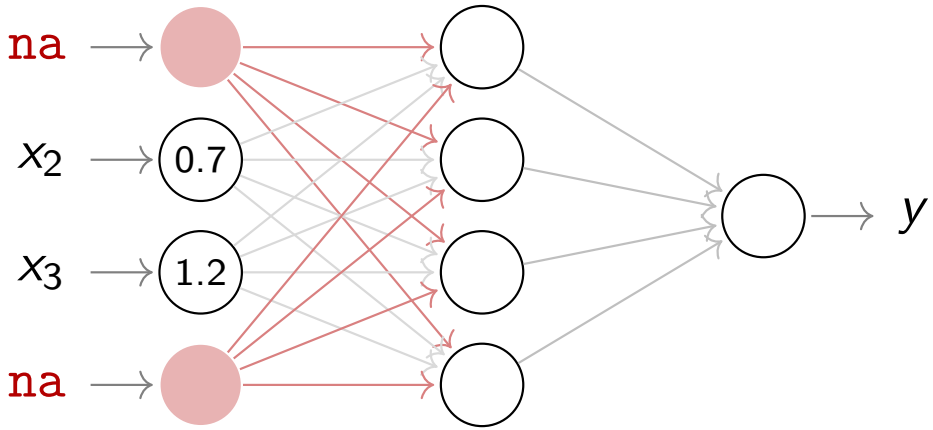


Challenges of supervised learning with missing values

- Supervised learning with missing values:
  - Scarcer literature (vs Inference, Imputation)
  - Requires handling arbitrary subsets of variables.

Samples

$$\begin{bmatrix} 1.1 & 0.2 & 0.5 & 1.3 \\ 0.1 & \texttt{na} & 0.2 & 1.4 \\ \texttt{na} & 0.7 & 1.2 & \texttt{na} \end{bmatrix}$$

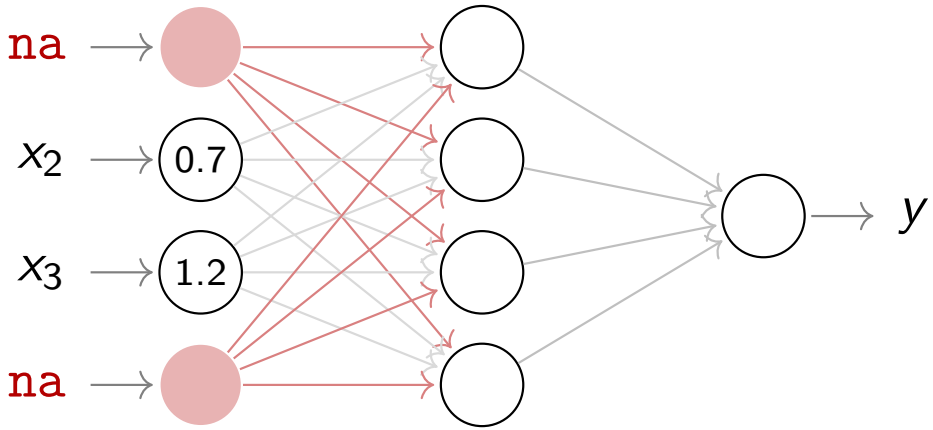


Challenges of supervised learning with missing values

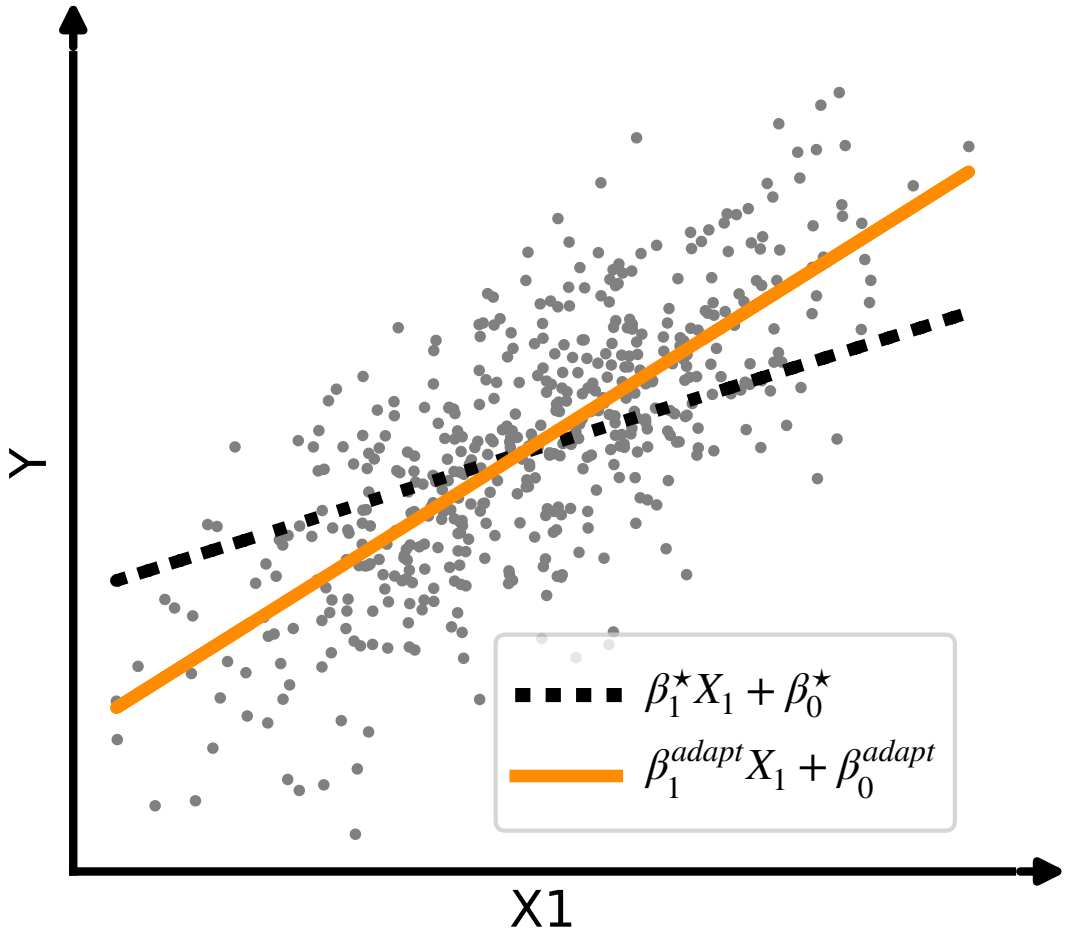
- Supervised learning with missing values:
  - Scarcer literature (vs Inference, Imputation)
  - Requires handling arbitrary subsets of variables.
  - Statistical challenge.
    Ex: $d = 50 \implies 2^{50} \approx 10^{15}$ possible missing data patterns.

Samples

$$\begin{bmatrix} 1.1 & 0.2 & 0.5 & 1.3 \\ 0.1 & \texttt{na} & 0.2 & 1.4 \\ \texttt{na} & 0.7 & 1.2 & \texttt{na} \end{bmatrix}$$



The case of linear regression with missing values: intuition



$$Y = \beta_1^\star X_1 + \beta_2^\star X_2 + \beta_0^\star$$

$$\text{cor}(X_1, X_2) = 0.5.$$

If X_2 is missing, the coefficient of X_1 should **compensate for the missingness of X_2**.

For a general problem in d dimensions, there are 2^d **possible missing data patterns**, and thus 2^d slopes to estimate.

The case of linear regression with missing values: intuition

Linear model: $Y = \beta_1^\star X_1 + \beta_2^\star X_2 + \beta_0^\star$

Suppose $(X_1, X_2) \sim \mathcal{N}(\mu, \Sigma)$.

Suppose only X_1 is observed.

Then the best prediction model on the observed data is:

$$\mathbb{E}\left[Y \mid X_1\right] = \beta_1^\star X_1 + \beta_2^\star \mathbb{E}\left[X_2 \mid X_1\right] + \beta_0^\star \tag{1}$$

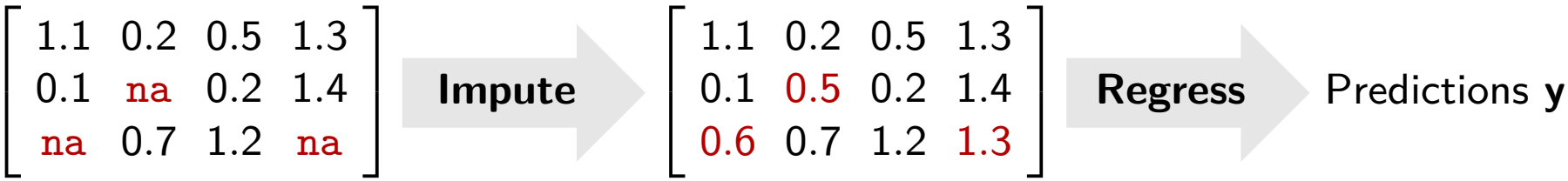
$$= \beta_1^\star X_1 + \beta_2^\star \left(\mu_2 + \frac{\Sigma_{2,1}}{\Sigma_{1,1}}\left(X_1 - \mu_1\right)\right) + \beta_0^\star \tag{2}$$

$$= \left(\beta_1^\star + \beta_2^\star \frac{\Sigma_{2,1}}{\Sigma_{1,1}}\right) X_1 + \left(\beta_2^\star \mu_2 - \beta_2^\star \frac{\Sigma_{2,1}}{\Sigma_{1,1}} \mu_1 + \beta_0^\star\right) \tag{3}$$

$$= \beta_1^{adapt} X_1 + \beta_0^{adapt} \tag{4}$$

How do people usually handle missing values?

Widespread current practice: Impute-then-Regress (or classify)



Important note: The imputation should be learned on the **train set only**, before being applied to the train and test sets.

A review of imputation strategies

Imputation

- i.e. filling in with likely values.

$$\begin{pmatrix} 1.2 & \texttt{na} & 0.2 & -0.4 \\ -0.3 & \texttt{na} & \texttt{na} & -0.9 \\ \texttt{na} & 0.4 & \texttt{na} & \texttt{na} \\ -2.8 & 1.9 & 1.6 & 2.2 \\ \texttt{na} & 1.7 & -2.4 & \texttt{na} \end{pmatrix}$$

- Naive imputations (e.g. constants).
- k-Nearest Neighbors.
- Likelihood-based approaches (EM).
- Multiple imputation by chained equation (MICE).
- Matrix factorization.
- Many recent approaches based on GANs, VAEs, GNNS, optimal transport, ...

Some simple imputation techniques

- **Univariate imputations**:
  - by a constant (e.g. 0).
  - by the mean or the median (for continuous features).
  - by the mode (for categorical variables).
- **k-Nearest Neighbors imputation**:
  - Nearest neighbours found based on observed features only.
  - Imputation using the mean value of the nearest neighbours.
- **Implementation in Python**:
  - Scikit-learn's `SimpleImputer` and `KNNImputer`.

Likelihood-based approach - 1/2

- We need to assume a **parametric model** for the probability density of X, with parameters θ:

$$X \sim p_\theta(X).$$

  For example, we can assume the data follows a Gaussian distribution with parameters $\theta = (\mu, \Sigma)$.

- With complete data, the maximum likelihood estimator (MLE) $\hat{\theta}$ is given by:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} p_\theta(X_i)$$

- In the presence of missing data, under **MAR** hypothesis, we showed that:
  - the missingness mechanism can be **ignored** and that consequently:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} p_\theta(X_{i,obs})$$

  - $\hat{\theta}$ can be obtained using an **EM algorithm**.

Likelihood-based approach - 2/2

- Once $\hat{\theta}$ is obtained, it is possible to **impute using** $p_{\hat{\theta}}(X_{mis}|X_{obs})$.
  - deterministically using the conditional expectation $\mathbb{E}\left[X_{mis}|X_{obs}\right]$.
  - or stochastically with random draws from $p_{\hat{\theta}}(X_{mis}|X_{obs})$.

- For p_θ, the **multivariate Gaussian distribution** is often used. It has the advantage of being analytically tractable.

  Indeed, if $p_\theta(X) = \mathcal{N}\left(X; \mu, \Sigma\right)$, the conditional Gaussian formula tells us that $p_\theta(X_{mis}|X_{obs})$ is also Gaussian with mean and covariance:

$$\mu_{mis|obs} = \mu_{mis} + \Sigma_{mis,obs}\Sigma_{obs}^{-1}\left(X_{obs} - \mu_{obs}\right) \tag{5}$$

$$\Sigma_{mis|obs} = \Sigma_{mis,mis} - \Sigma_{mis,obs}\Sigma_{obs}^{-1}\Sigma_{obs,mis} \tag{6}$$

- Implementation:
  - Python: no widely accepted package.
  - R: package `norm` (assumes a Gaussian distribution).

MICE - 1/2

- Standard multivariate distributions (e.g. Gaussian) do not always provide a good fit for real data.
- **Idea**: Specify a **conditional distribution for each variable** X_j depending on all others.
- MICE is an **iterative algorithm**:
  1. For each $j \in \left[1, \ldots, J\right]$, specify :

$$p_{\theta_j}(X_j | X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_J)$$

  2. Create initial approximate imputations using a simple procedure.
  3. For a number of iterations T:
     For each variable j in turn:
     ✓Update θ_j .
     ✓Update imputations with random draws from $p_{\theta_j}(X_j | \ldots)$.
- We can also define a deterministic version of MICE.

MICE - 1/2

- Standard multivariate distributions (e.g. Gaussian) do not always provide a good fit for real data.
- **Idea**: Specify a **conditional distribution for each variable** X_j depending on all others.
- MICE is an **iterative algorithm**:
  1. For each $j \in \left[1, \ldots, J\right]$, specify a regression model:

$$X_j = f_{\theta_j}(X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_J).$$

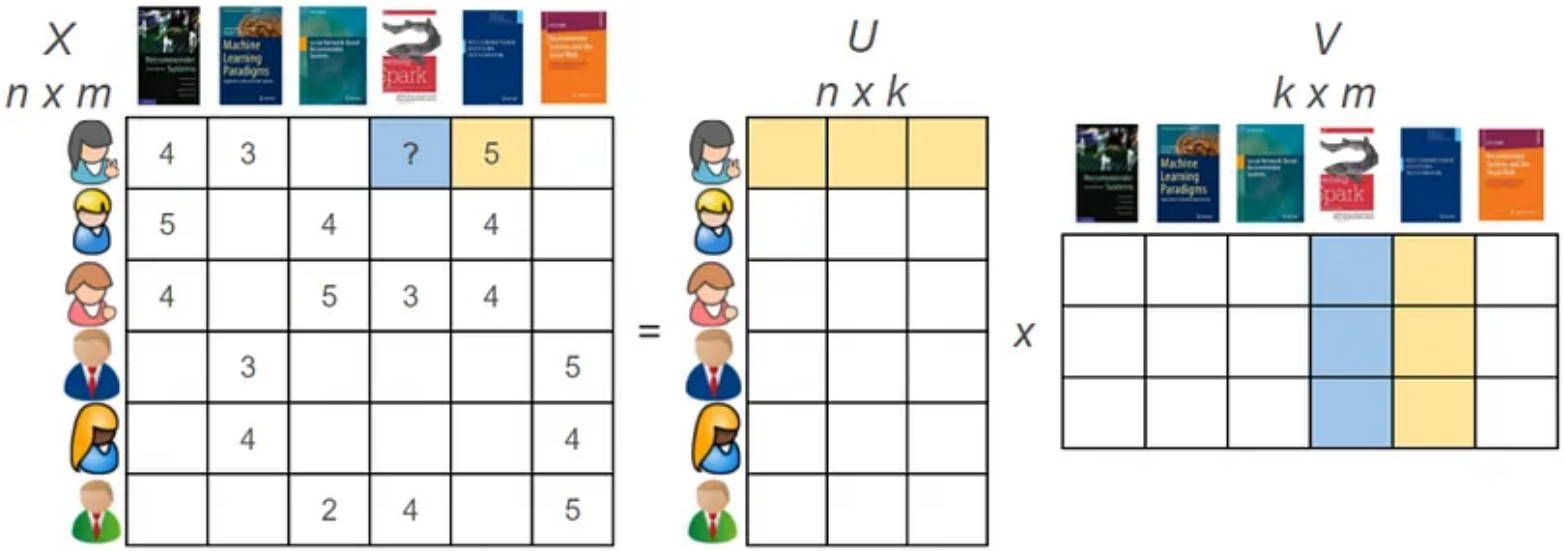
  2. Create initial approximate imputations using a simple procedure.
  3. For a number of iterations T:
     For each variable j in turn:
     ✓ Learn θ_j.
     ✓Update imputations with predictions $f_{\theta_j}(X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_J)$.
- We can also define a deterministic version of MICE.

MICE - 2/2

- Advantages:
  - Can handle any type of variable (continuous, binary, categorical). Just define the conditional distribution/regression function accordingly.
  - State-of-the-art performances, notably with `missforest`, a deterministic version where random forest are used for regressions.
- Inconveniences:
  - Questions about the convergence.
  - Computationally expensive.
- Implementation:
  - In Python: Scikit-learn's `IterativeImputer`.
  - in R: `mice`, `missforest` (+ other options).

Imputation by Matrix Factorization

- **Idea**: Find latent factors to reconstruct well the observed values of a matrix.



- Many techniques coming from the literature on Recommender systems.
- Many algorithms:
  - PCA with missing values (EM algorithm again)
  - Iterated SVD
  - Alternated Least squares
  - Funk SVD (stochastic gradient descent on the reconstruction error)
  - ...

A jungle of recent imputation methods

Missing Data Imputation using Optimal Transport

MIWAE: Deep Generative Modelling and Imputation of Incomplete Data Sets

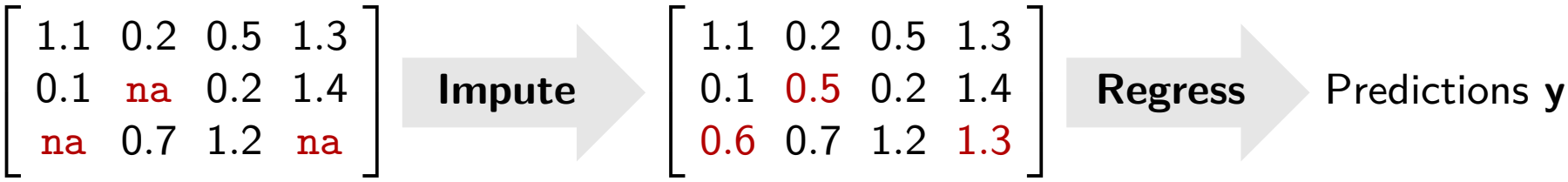
Handling Missing Data with Graph Representation Learning

Missing Value Imputation for Mixed Data via Gaussian Copula

GAIN: Missing Data Imputation using Generative Adversarial Nets

and many others!

Widespread current practice: Impute-then-Regress (or classify)



Question:
Should we impute well to predict well?

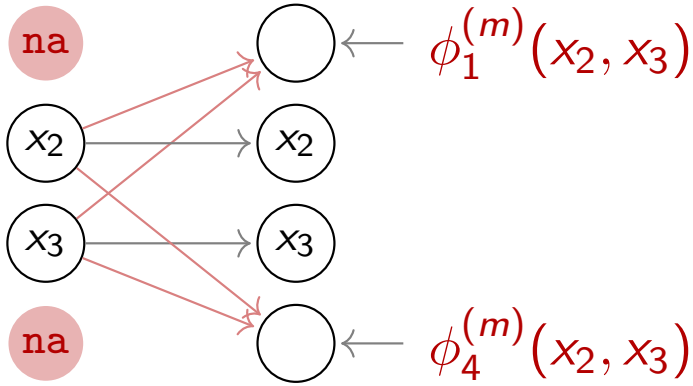
What do current theoretical foundations tell us?

Impute-then-Regress procedures

- Define Impute-then-Regress procedures as functions of the form:

$$g \circ \Phi, \text{ where } \Phi \in \mathcal{F}^I, \; g : \mathbb{R}^d \mapsto \mathbb{R}.$$

where imputation functions $\Phi \in \mathcal{F}^I$ are of the form:



Questions

- *Can Impute-then-Regress procedures be Bayes optimal?*
- *How should we choose the imputation function?*
- *What if the data is Missing Not At Random (MNAR)?*

Can Impute-then-Regress procedures be Bayes optimal?

Can Impute-then-Regress procedures be Bayes optimal?

Theorem (Bayes optimality of Impute-then-Regress procedures $*$)

For almost all imputation functions, and for all missingness mechanisms, a universally consistent algorithm trained on the imputed data is Bayes consistent.

$*$ Informal version, from *Le Morvan et al, What's a good imputation to predict with missing values?, Neurips 2021*

Can Impute-then-Regress procedures be Bayes optimal?

Theorem (Bayes optimality of Impute-then-Regress procedures $*$)

For almost all imputation functions, and for all missingness mechanisms, a universally consistent algorithm trained on the imputed data is Bayes consistent.

$*$ Informal version, from *Le Morvan et al, What's a good imputation to predict with missing values?, Neurips 2021*

Questions

- *Can Impute-then-Regress procedures be Bayes optimal?*
- *How should we choose the imputation function?*
- *What if the data is Missing Not At Random (MNAR)?*

Can Impute-then-Regress procedures be Bayes optimal?

Theorem (Bayes optimality of Impute-then-Regress procedures $*$)

For almost all imputation functions, and for all missingness mechanisms, a universally consistent algorithm trained on the imputed data is Bayes consistent.

$*$ Informal version, from *Le Morvan et al, What's a good imputation to predict with missing values?, Neurips 2021*

Questions

- *Can Impute-then-Regress procedures be Bayes optimal?*

  ✓ Yes! In fact, they almost always are.

- *How should we choose the imputation function?*

- *What if the data is Missing Not At Random (MNAR)?*

Can Impute-then-Regress procedures be Bayes optimal?

Theorem (Bayes optimality of Impute-then-Regress procedures $*$)

For almost all imputation functions, and for all missingness mechanisms, a universally consistent algorithm trained on the imputed data is Bayes consistent.

$*$ Informal version, from *Le Morvan et al, What's a good imputation to predict with missing values?, Neurips 2021*

Questions

- *Can Impute-then-Regress procedures be Bayes optimal?*

  ✓ Yes! In fact, they almost always are.

- *How should we choose the imputation function?*

  ✓ How you want! (almost) Constant imputations work.

- *What if the data is Missing Not At Random (MNAR)?*

Can Impute-then-Regress procedures be Bayes optimal?

Theorem (Bayes optimality of Impute-then-Regress procedures $*$)

For almost all imputation functions, and for all missingness mechanisms, a universally consistent algorithm trained on the imputed data is Bayes consistent.

$*$ Informal version, from *Le Morvan et al, What's a good imputation to predict with missing values?, Neurips 2021*

Questions

- *Can Impute-then-Regress procedures be Bayes optimal?*

  ✓ Yes! In fact, they almost always are.

- *How should we choose the imputation function?*

  ✓ How you want! (almost) Constant imputations work.

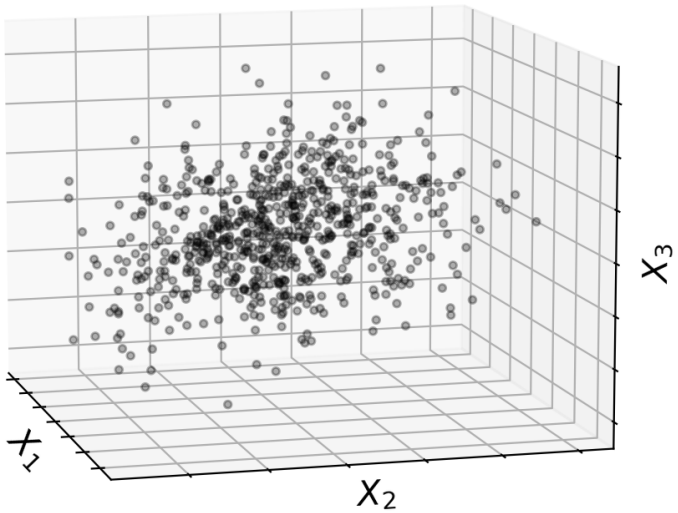
- *What if the data is Missing Not At Random (MNAR)?*

  ✓ The missingness mechanism does not matter.

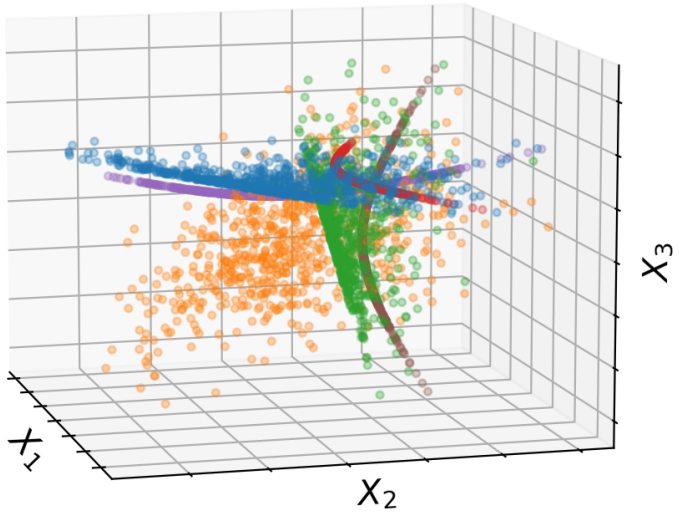
Sketch of the proof: arguments from differential topology

Sketch of the proof:

1. All data points with a missing data pattern m are mapped to a manifold $\mathcal{M}^{(m)}$ of dimension $|obs(m)|$ (Preimage Theorem).



Complete data

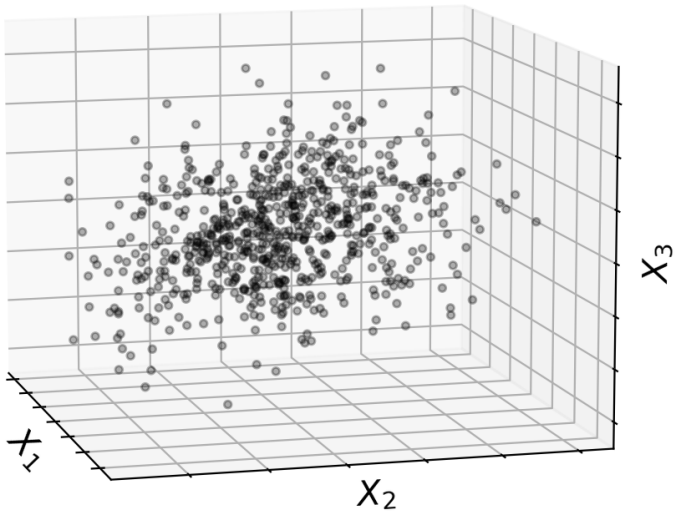


Imputed data (manifolds)

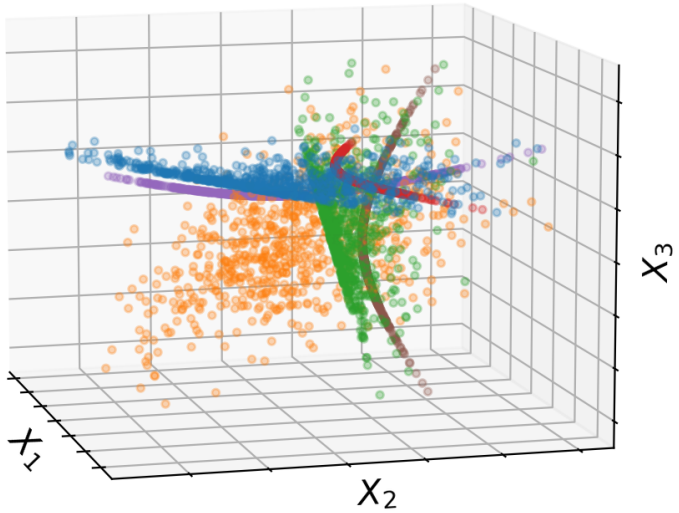
Sketch of the proof: arguments from differential topology

Sketch of the proof:

1. All data points with a missing data pattern m are mapped to a manifold $\mathcal{M}^{(m)}$ of dimension $|obs(m)|$ (Preimage Theorem).
2. The missing data patterns of imputed data points can almost surely be de-identified (Thom transversality Theorem).



Complete data

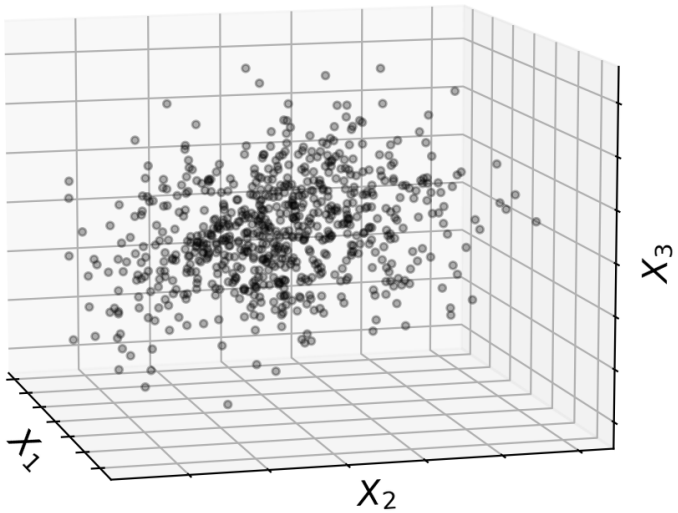


Imputed data (manifolds)

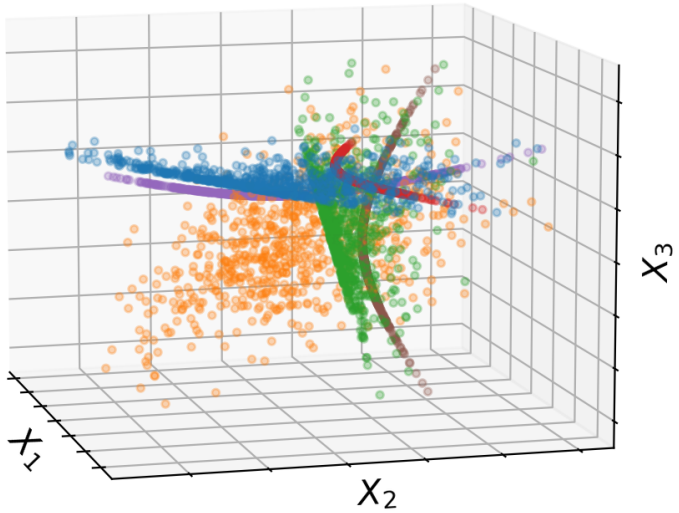
Sketch of the proof: arguments from differential topology

Sketch of the proof:

1. All data points with a missing data pattern m are mapped to a manifold $\mathcal{M}^{(m)}$ of dimension $|obs(m)|$ (Preimage Theorem).
2. The missing data patterns of imputed data points can almost surely be de-identified (Thom transversality Theorem).
3. Given 2), we can exhibit a $g^{\star}_{\Phi}$, which does not depend on m, and which for each manifold equals the Bayes predictor except on a set of measure 0.

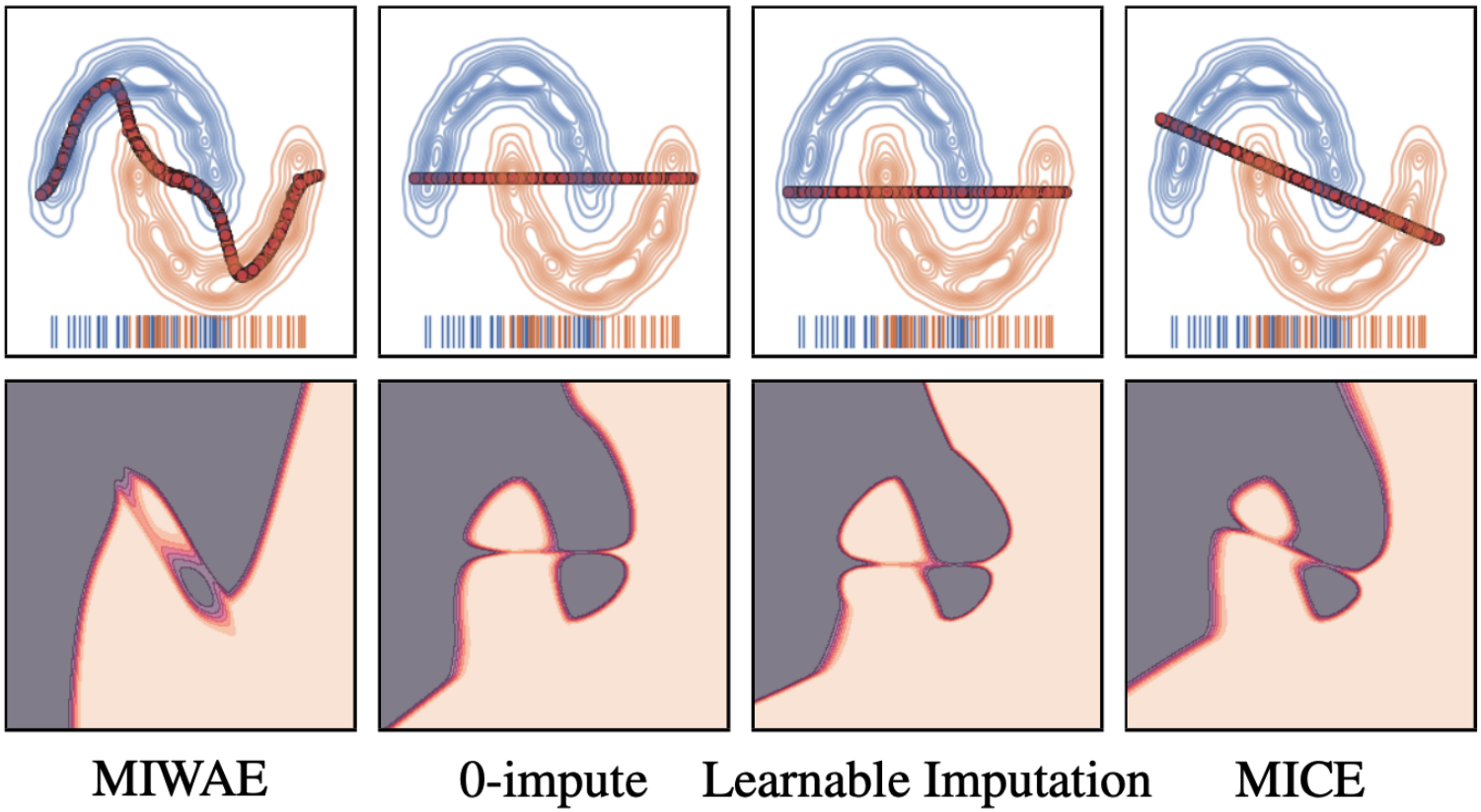


Complete data



Imputed data (manifolds)

Adaptation of the decision boundary to imputation manifolds



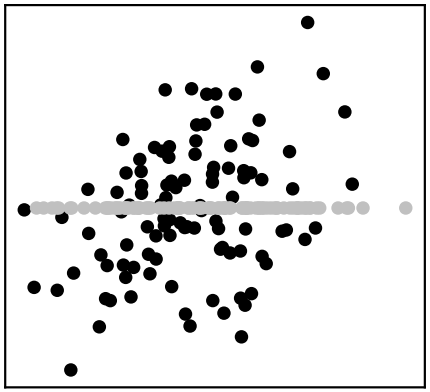
from Ipsen et al, How to deal with missing data in supervised deep learning?, ICLR 2022

Top row: data sensity and imputation manifold
Bottom row: Learned decision surface

So, what imputation to choose?

Inference ✗

Distorted distribution



mean imputation

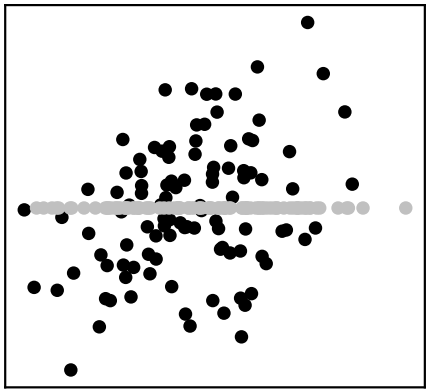
Supervised learning ✓

Bayes optimality

So, what imputation to choose?

Inference ✗

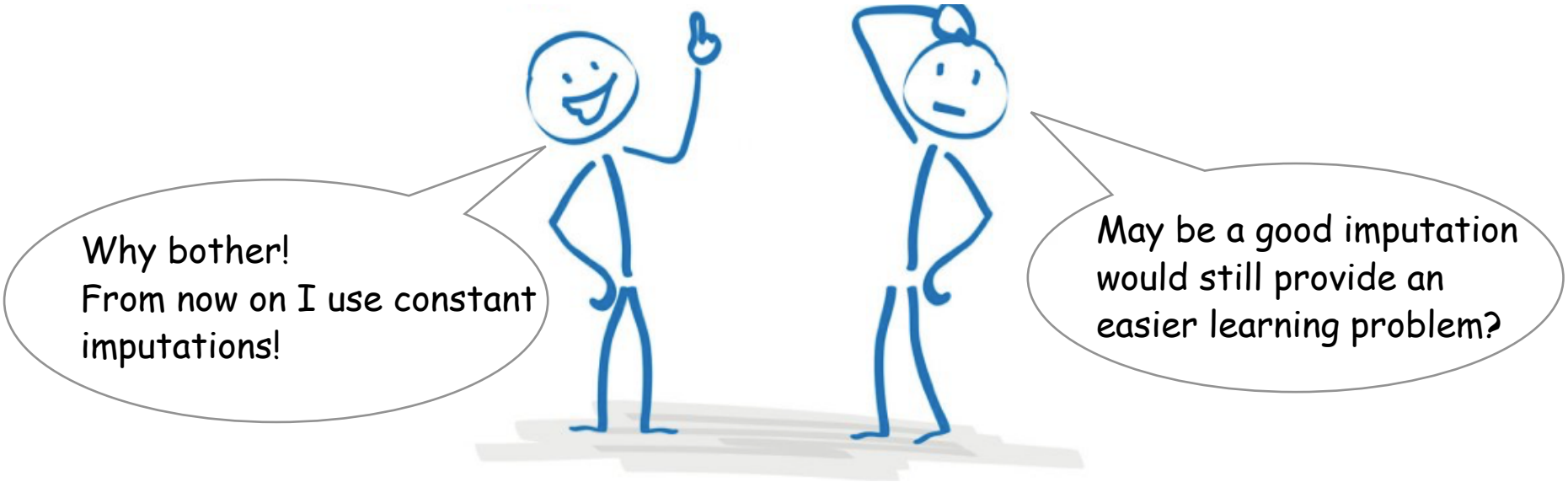
Distorted distribution



mean imputation

Supervised learning ✓

Bayes optimality



A tour of existing benchmarks.

A few experimental endeavors exist but most of them suffer from important limitations:

- Introducing missing values on a single variable chosen at random.
- Focusing on a specific model for prediction, for ex., a linear model.
- Drawing conclusions from 4 datasets.
- Performing imputation separately on the train and the test sets.
- No quantitative assessment of the relationship between imputation and prediction performances.

Many of these problems result from choices to reduce the high computational cost of the experiments.

Few works draw insights into when imputation could improve predictions.

Question
Should we impute well to predict well?

We quantify the change in predictive performance resulting from a gain in imputation accuracy across controlled experimental settings.

Imputation for prediction: beware of diminishing returns. ICLR 2025

Experimental design.

Our goal:
- ✓ Computing correlations between imputation and prediction quality.
- ✓ Determining if/when imputation improve predictions.

We strive to set up experiments in order to gather useful insights :

- **Impact of the downstream prediction model**: Assessing multiple prediction models, each combined with multiple imputation strategies.
- **Impact of the mask**: Each combination is tried with and without appending the mask as input.
- **Impact of the outcome non-linearity**: Each experiment is run on the real data + a semi-synthetic version with a linear outcome.
- **Generalization across datasets**: Experiments on 19 datasets.

The 19 datasets.

dataset	d	n_train	n_test
house_16H	16	18185	2274
cpu_act	21	6553	820
elevators	16	13279	1661
wine_quality	11	5197	651
Brazilian_houses	8	8553	1070
house_sales	15	17290	2162
sulfur	6	8064	1009
Ailerons	33	11000	1375
Bike_Sharing_Demand	6	13903	1739
california	8	16512	2064
diamonds	6	43152	5394
fifa	5	14450	1807
houses	8	16512	2064
isolet	613	6237	781
medical_charges	3	50000	50000
MiamiHousing2016	13	11145	1394
nyc-taxi-green-dec-2016	9	50000	50000
pol	26	12000	1500
superconduct	79	17010	2127
year	90	50000	50000

A tabular benchmark suite from **?**.

We focus on a presumably favorable ground for better imputations to improve predictions:

- Regression.
- Numerical features only (no categorical features).
- MCAR missingness.

We use a semi-synthetic version of each dataset where Y is generated as a linear function of X.

Impute-then-Regress combinations

Prediction models:

- MLP: Multilayer perceptron baseline.
- SAINT: best deep tabular model representative according to **?**.
- XGBoost: best tree-model representative according to **?**.

Imputation strategies:

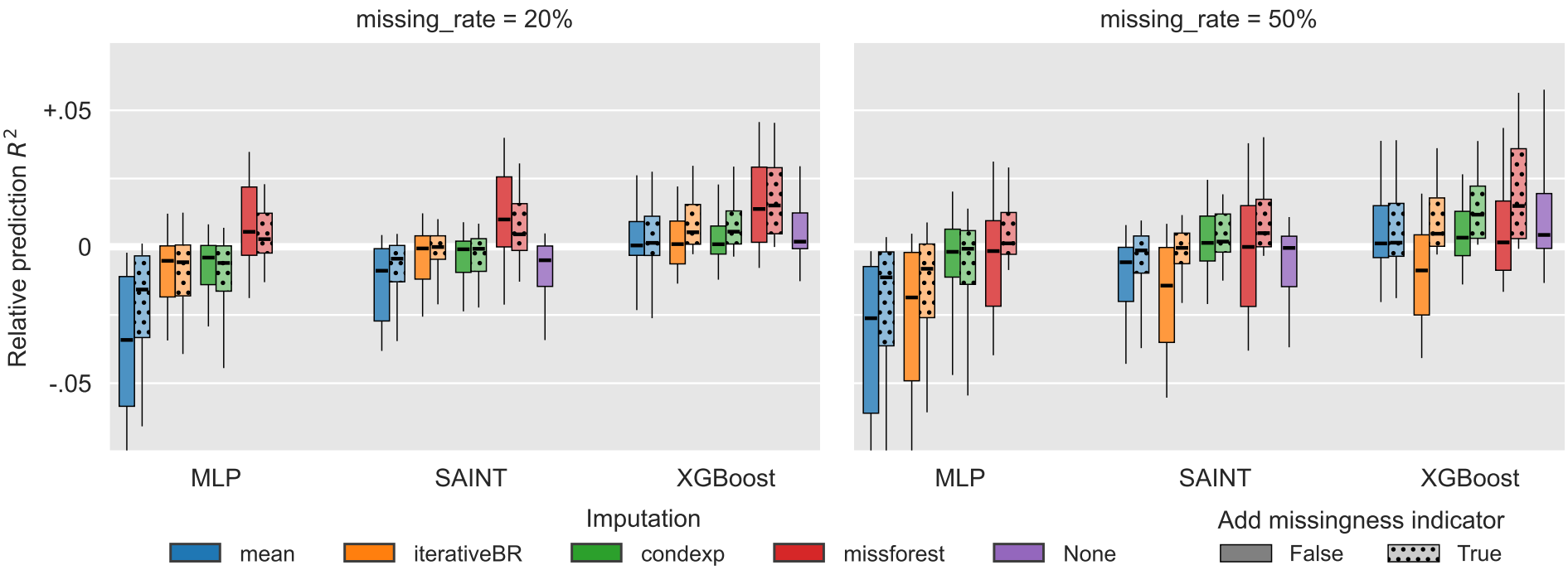
We look for a wide range of imputation qualities to calculate correlations.

- mean
- iterative imputation with a ridge model (MICE-like strategy)
- missforest
- condexp: Gaussian conditional expectation, with pairwise available case estimates of the mean and cov.

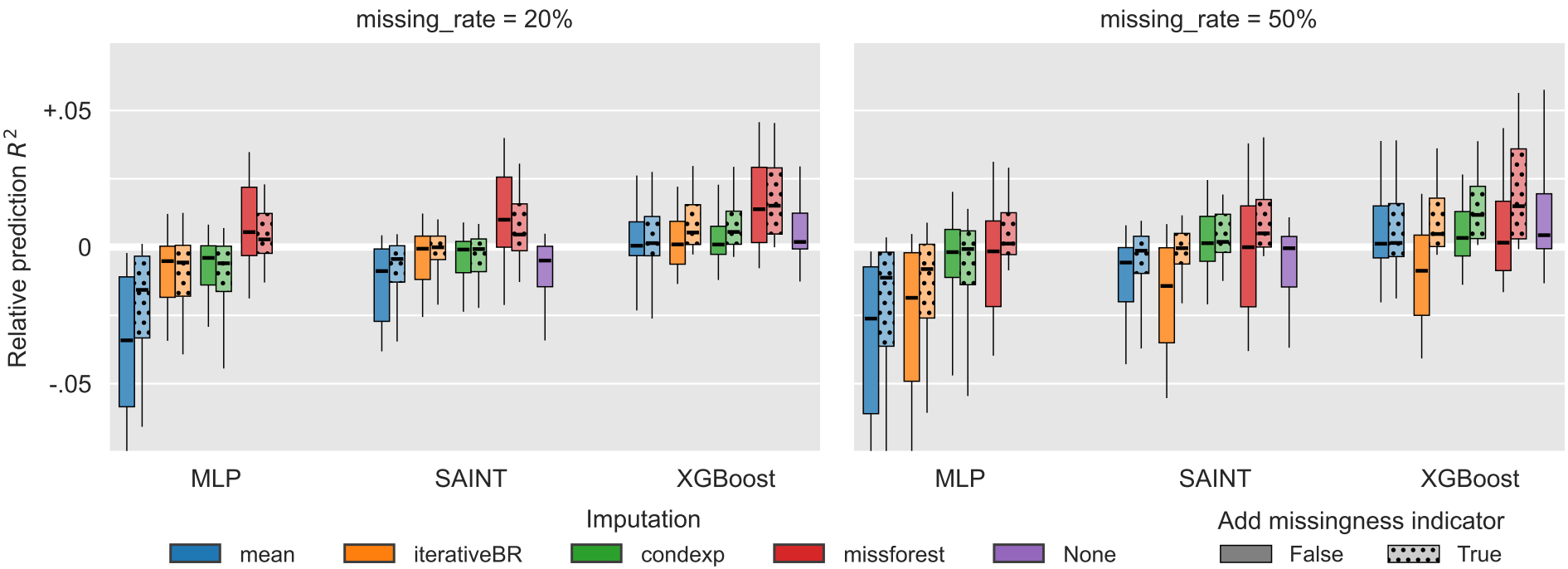
Computational load: 2.5 CPU years

- 26 Impute-then-Regress models (+/- the mask).
- 50 trials for hyperparameter tuning (with Optuna).
- 10 repetitions of each experiment.
- 19 datasets $\times 2$ outcomes (real + semi-synthetic linear outcome) $\times 3$ missing rates (20%, 50% and 70%) $\rightarrow$ 114 datasets.

The keys to good predictions with MCAR missingness.

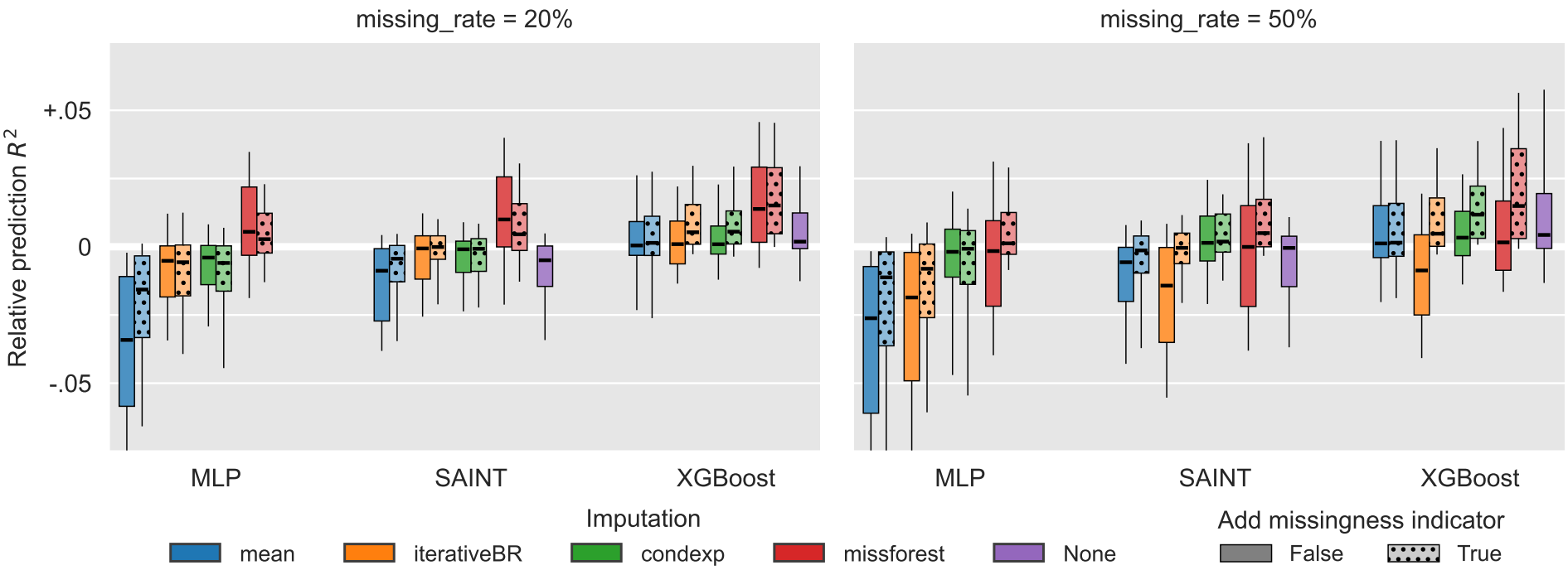


The keys to good predictions with MCAR missingness.



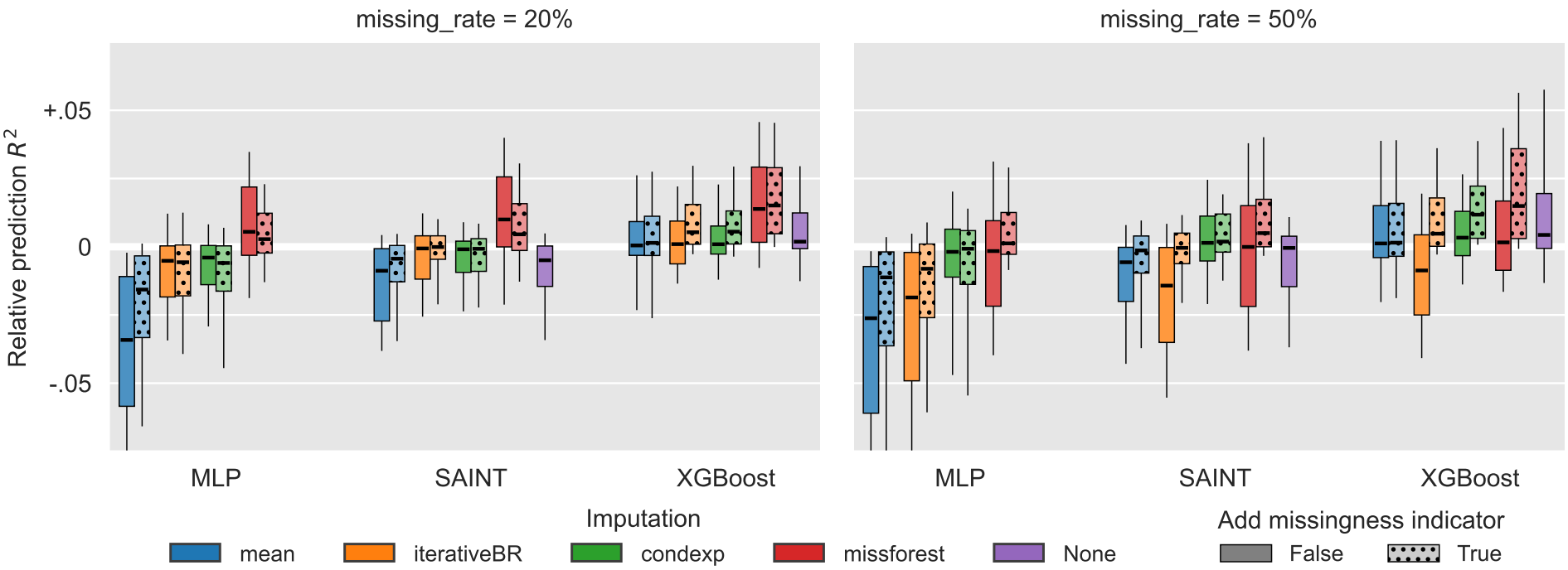
☞ more sophisticated imputers tend to improve prediction, *but*:

The keys to good predictions with MCAR missingness.



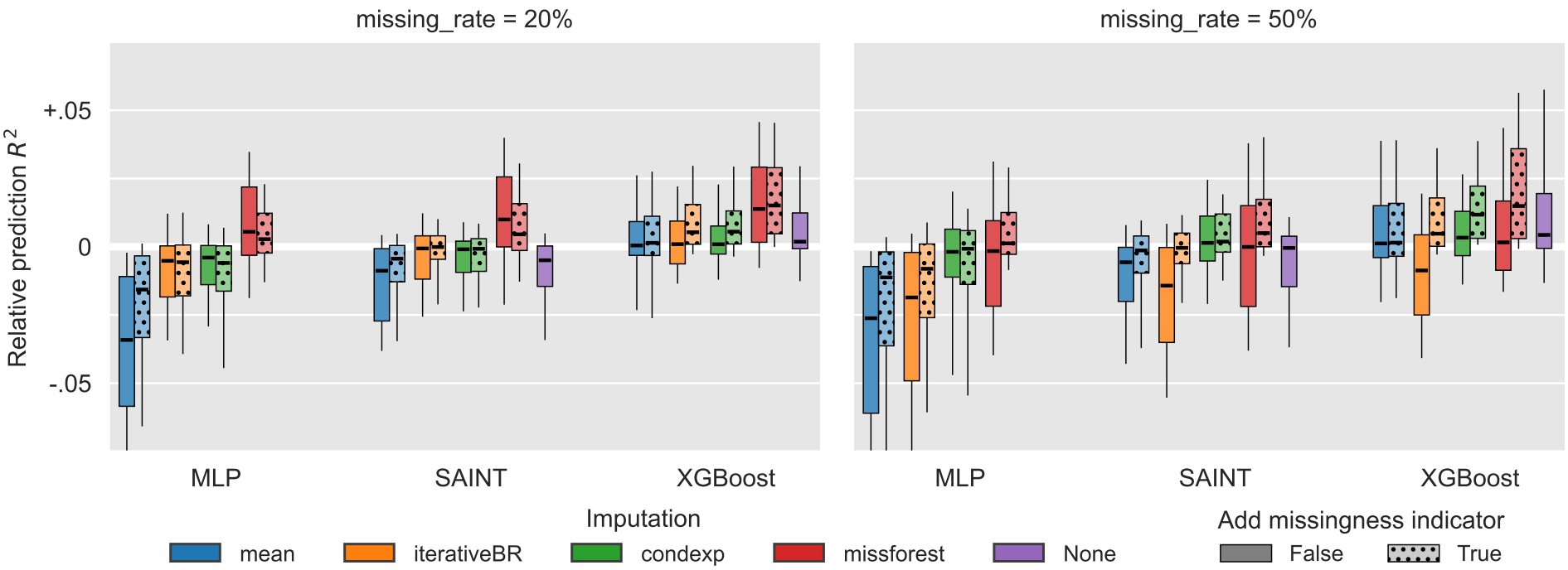
- ☞ more sophisticated imputers tend to improve prediction, *but*:
- ☞ using the missingness indicator decreases this effect,

The keys to good predictions with MCAR missingness.



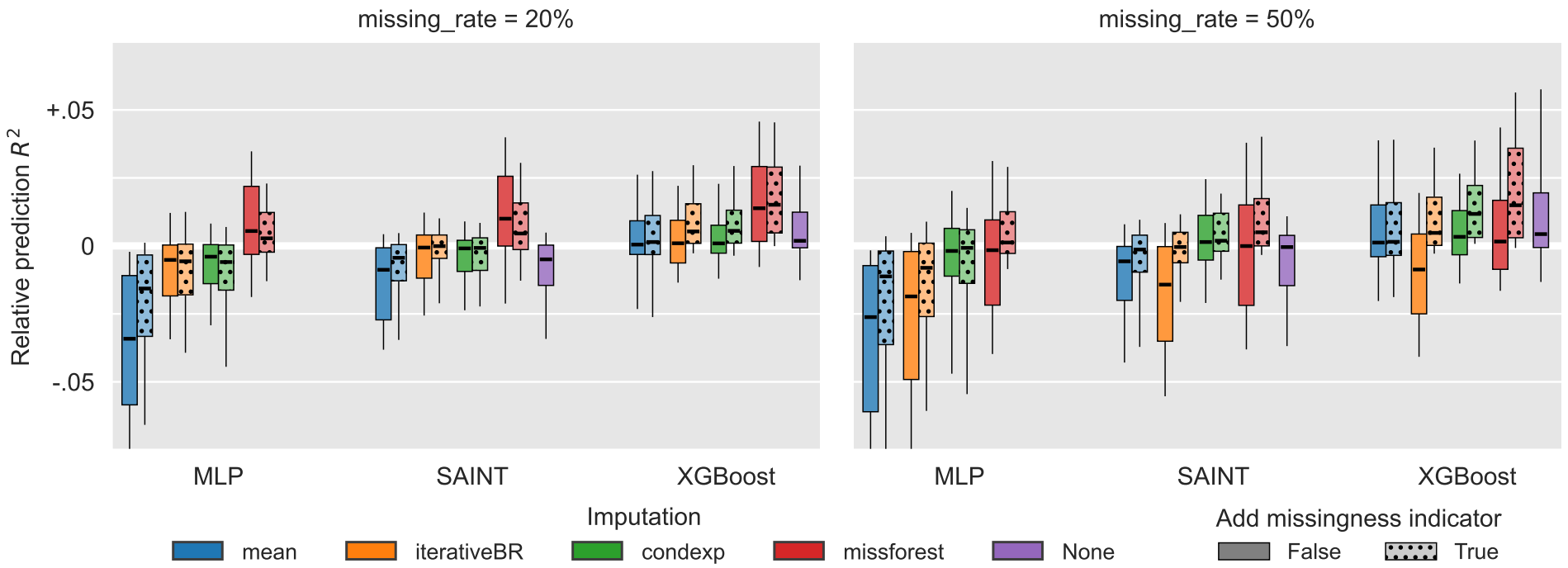
- ☞ more sophisticated imputers tend to improve prediction, *but*:
- ☞ using the missingness indicator decreases this effect,
- ☞ this effect is barely noticeable for the best predictor, XGBoost.

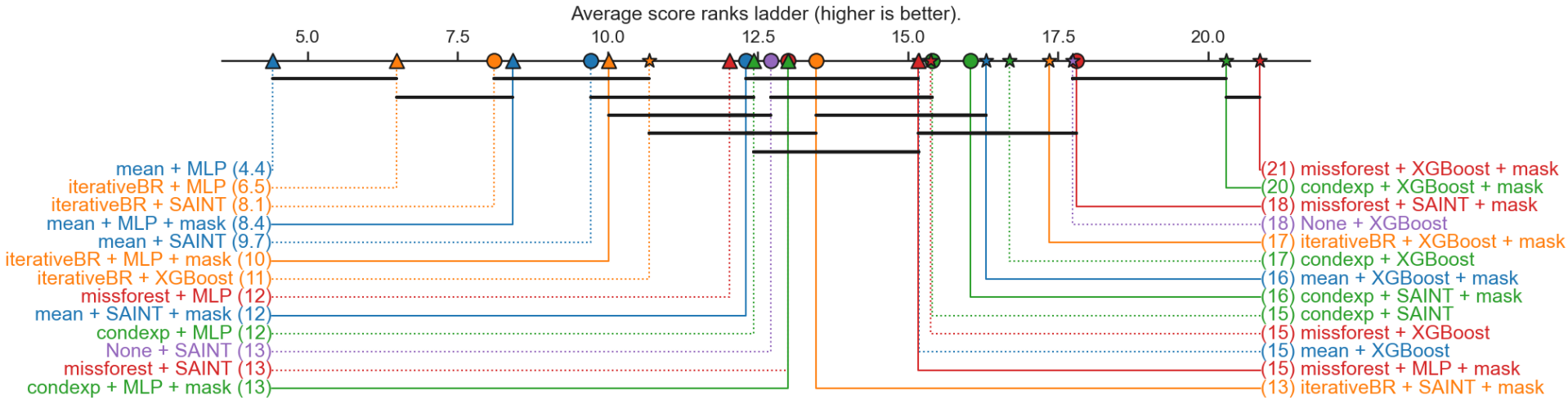
The keys to good predictions with MCAR missingness.



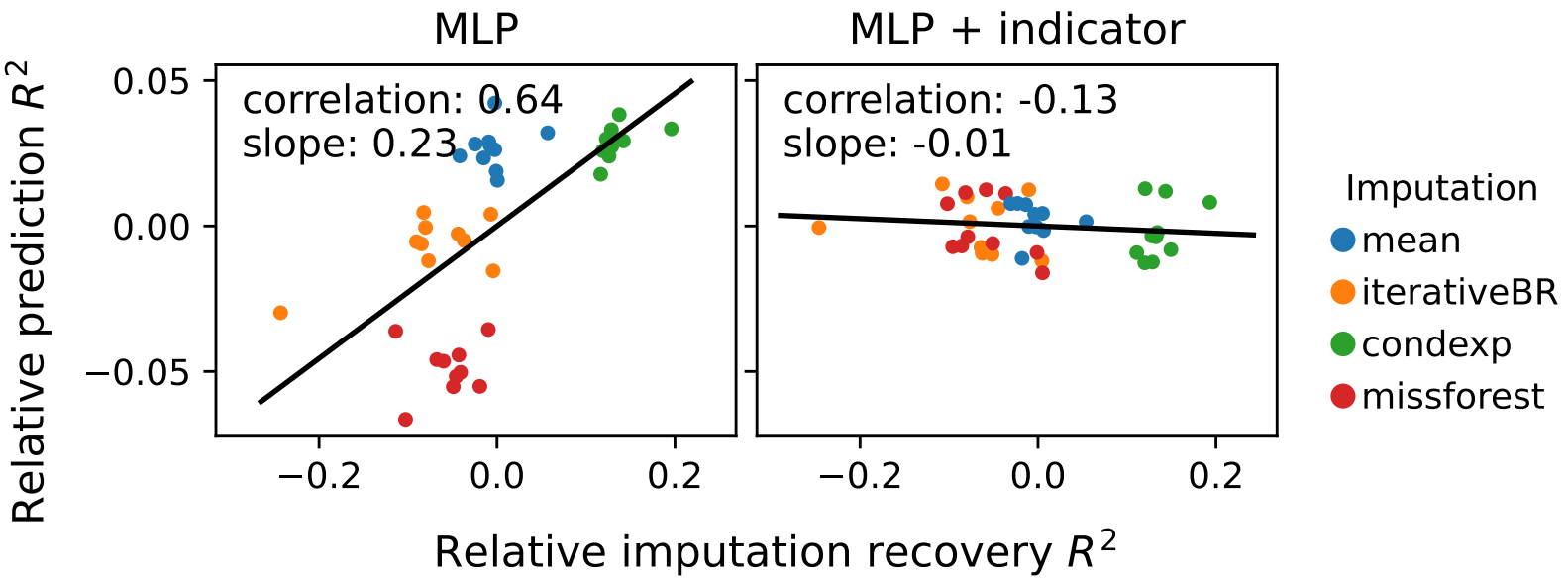
- ☞ more sophisticated imputers tend to improve prediction, *but*:
- ☞ using the missingness indicator decreases this effect,
- ☞ this effect is barely noticeable for the best predictor, XGBoost.
- ☞ No one-size-fits-all champion: high variance across datasets.

The keys to good predictions with MCAR missingness.



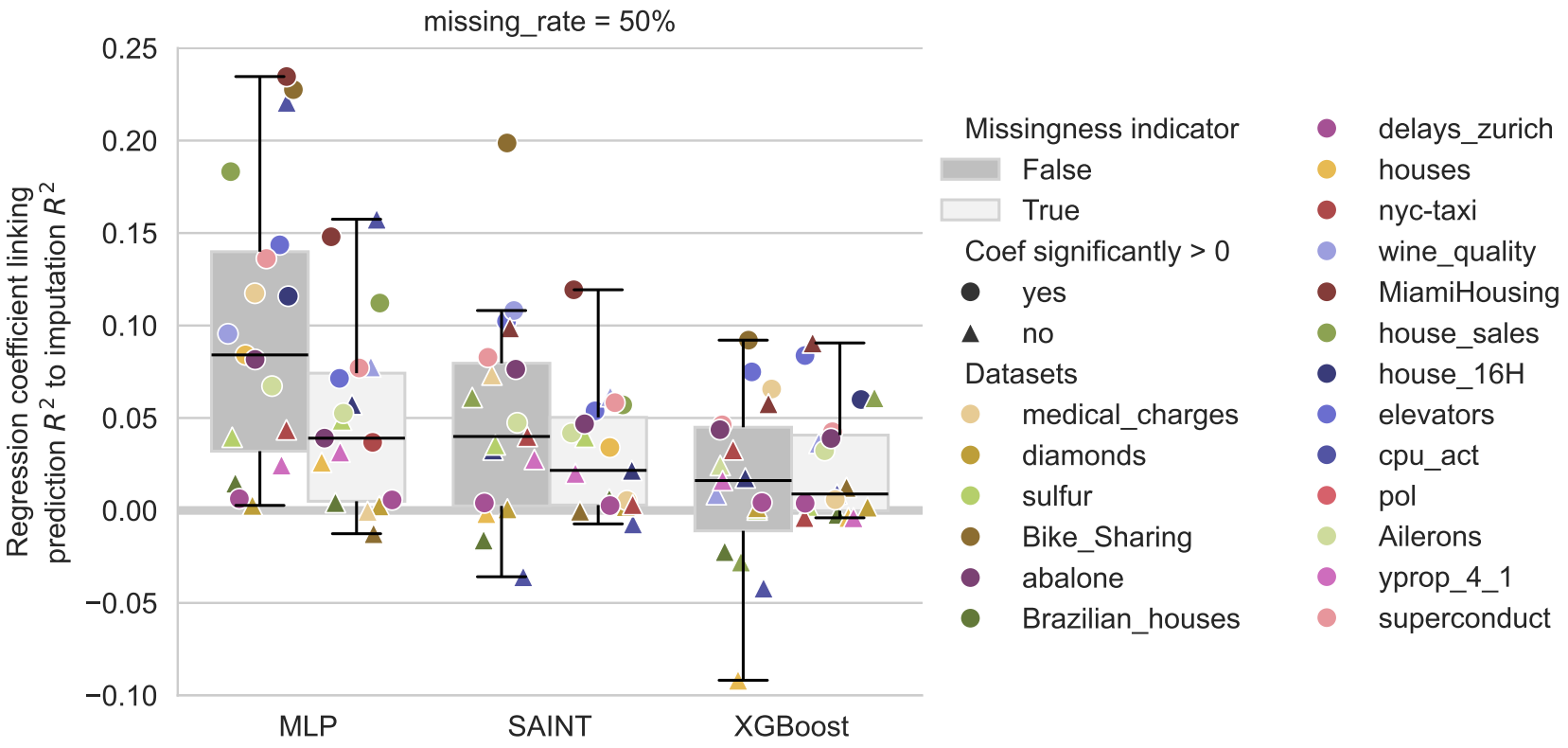


Quantifying the relationship between prediction R2 and imp. R2.

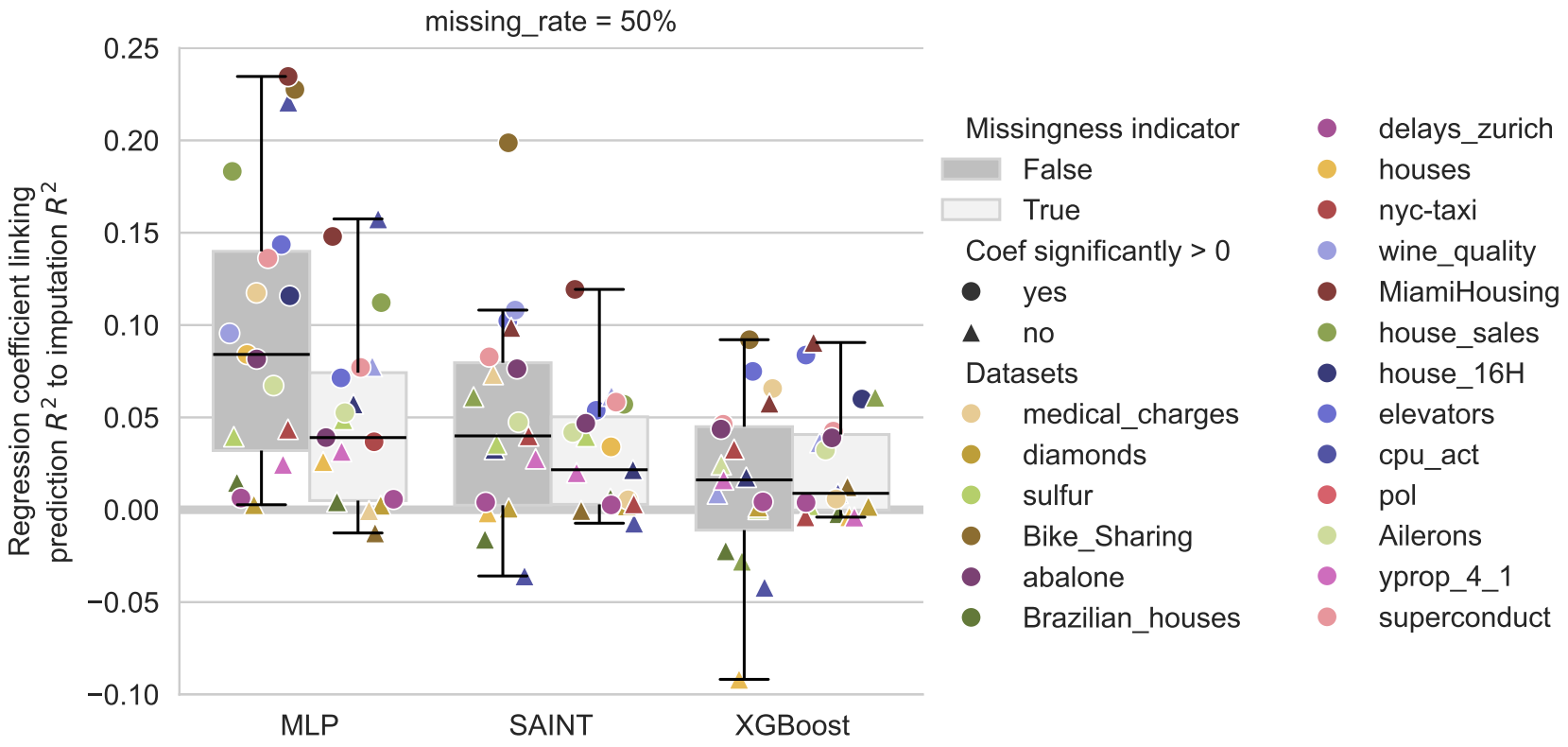


Example fit of prediction performance as a function of imputation accuracy, for the Bike_Sharing_Demand dataset and a missing rate of 50%.

Effect of the imputation quality on the prediction performance.

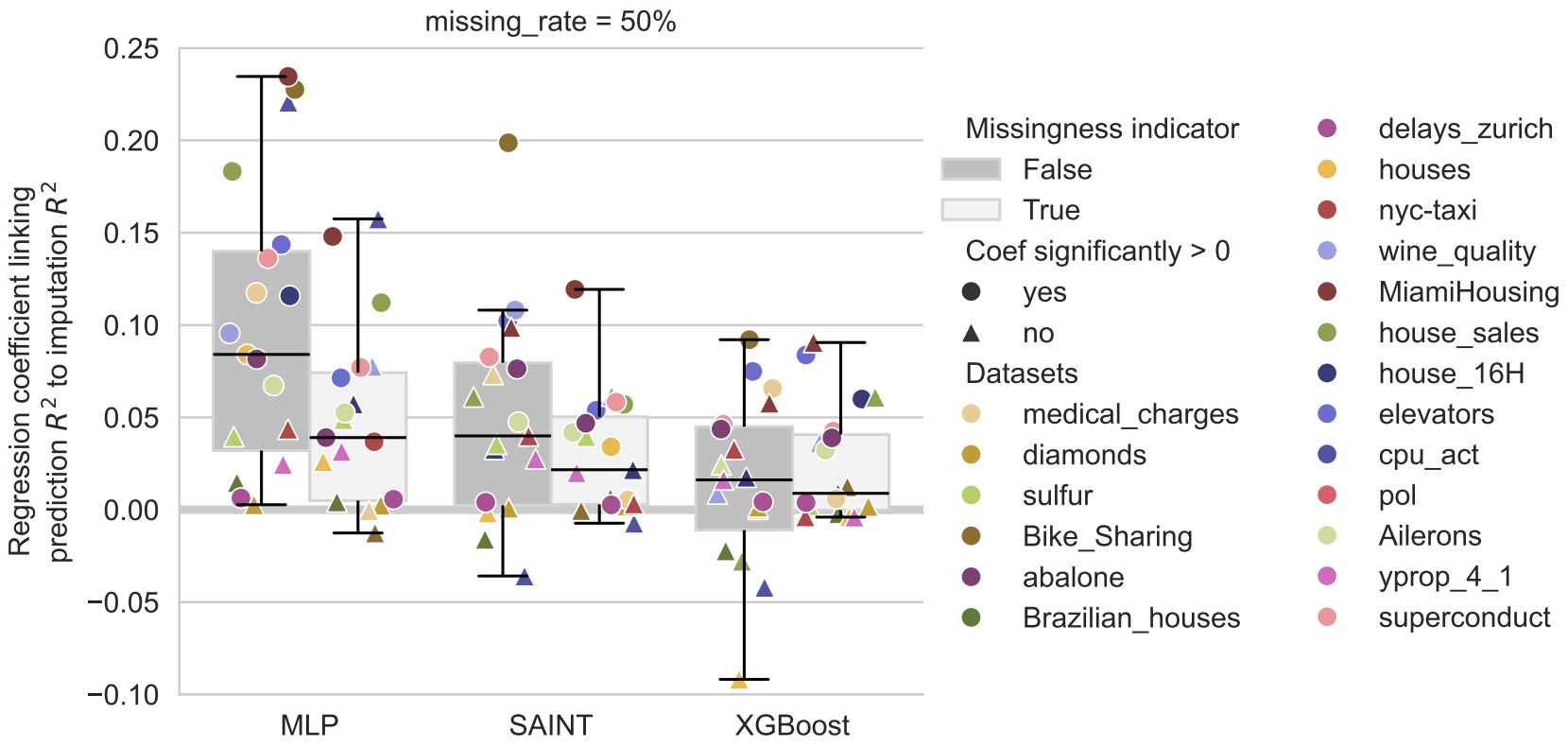


Effect of the imputation quality on the prediction performance.



☞ Gains in prediction R2 are 10% or less of the gains in imputation R2.

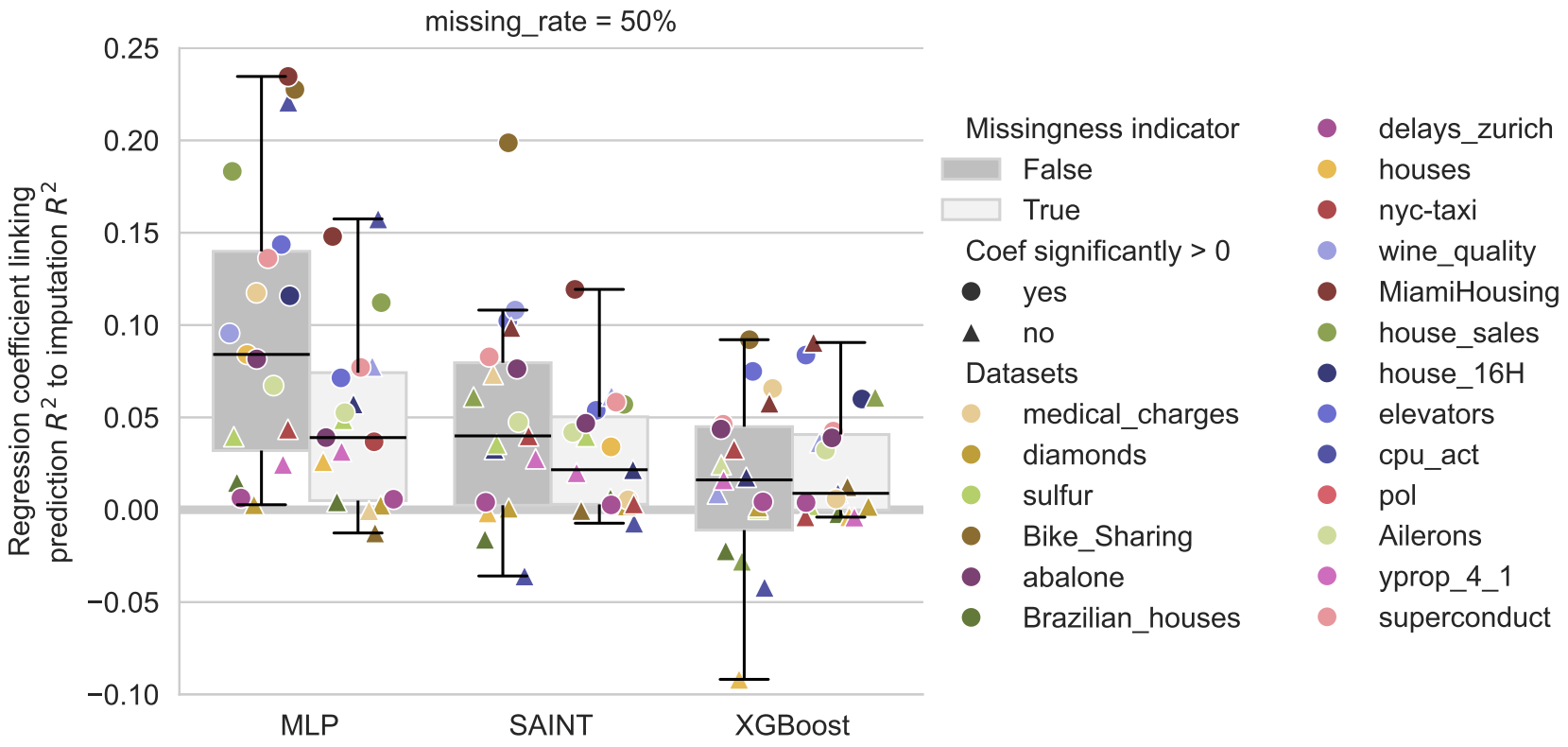
Effect of the imputation quality on the prediction performance.



☞ Gains in prediction R2 are 10% or less of the gains in imputation R2.

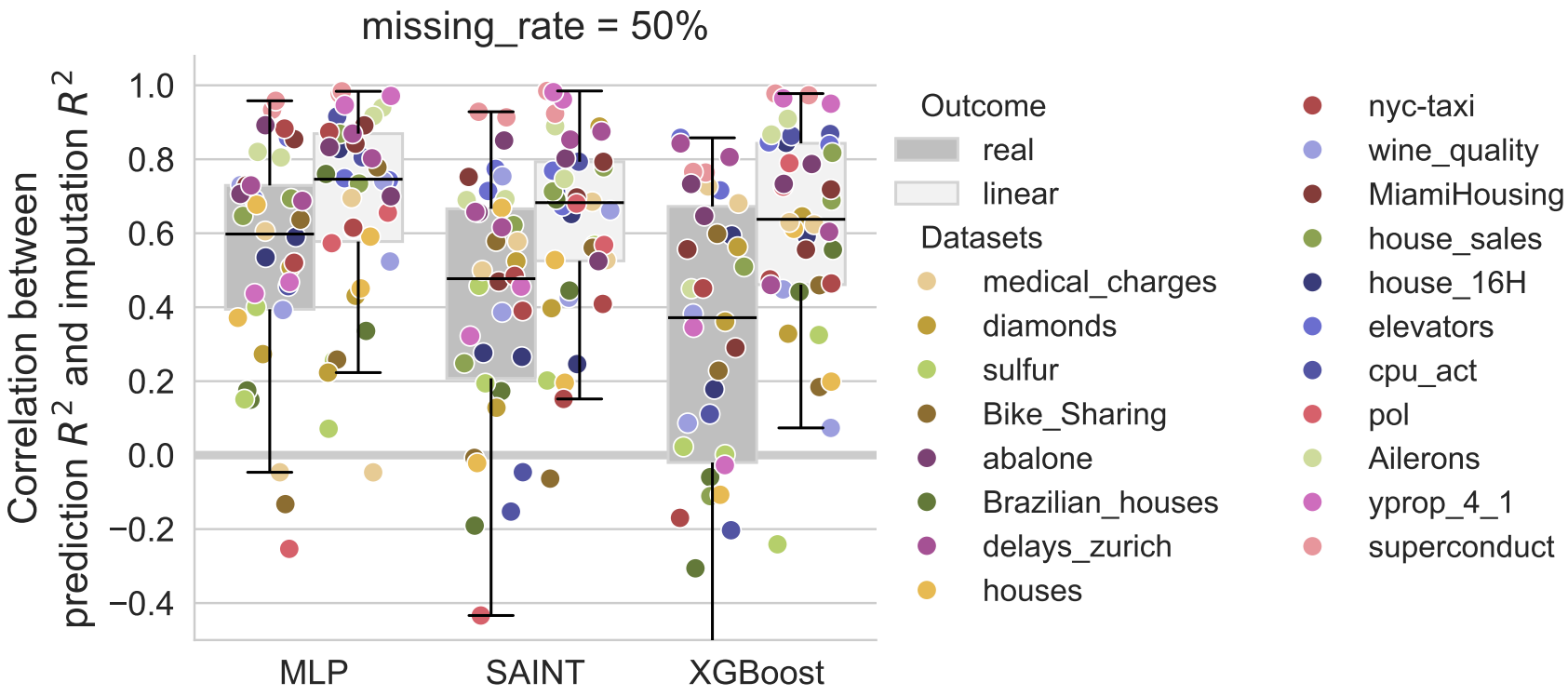
☞ Good imputations matter less when using the mask.

Effect of the imputation quality on the prediction performance.

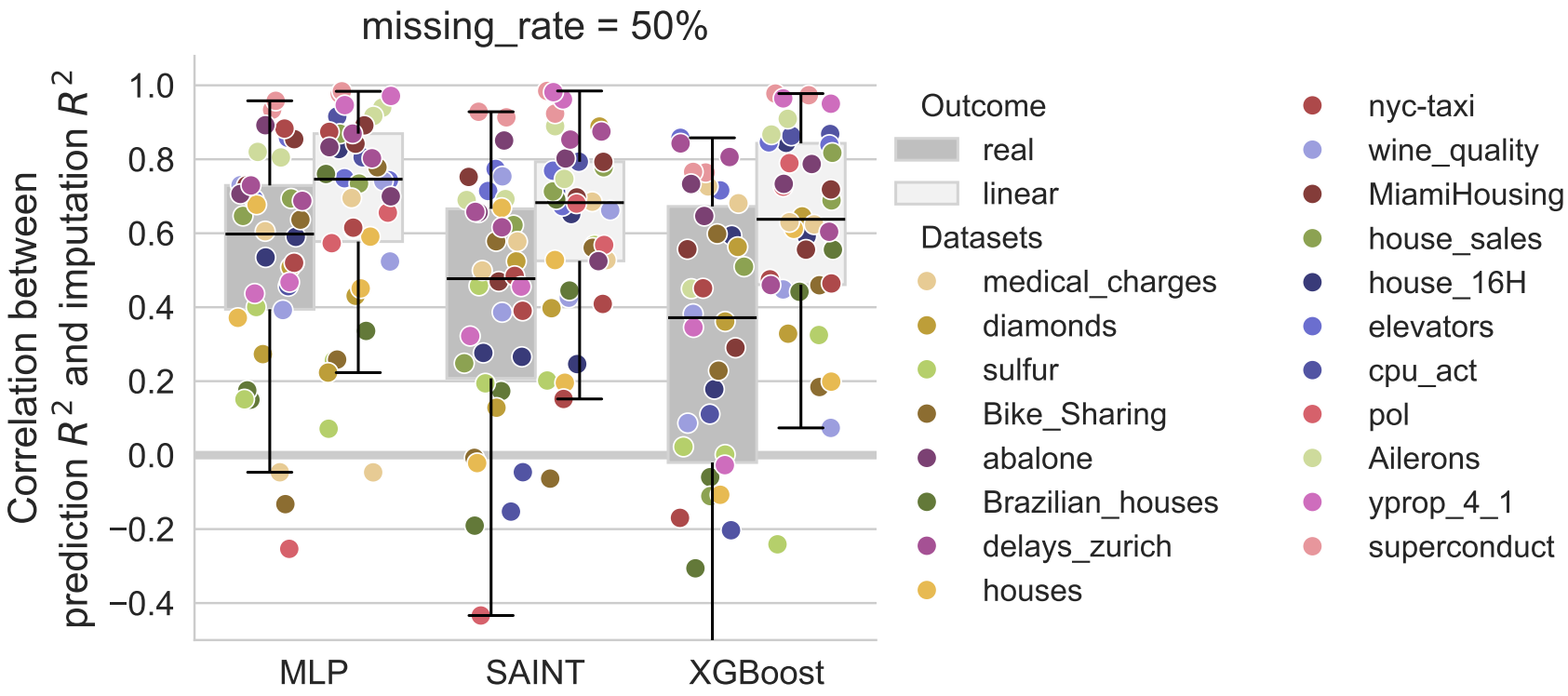


- ☞ Gains in prediction R2 are 10% or less of the gains in imputation R2.
- ☞ Good imputations matter less when using the mask.
- ☞ Good imputations matter less for more expressive models.

Partial correlation between imputation quality and performance

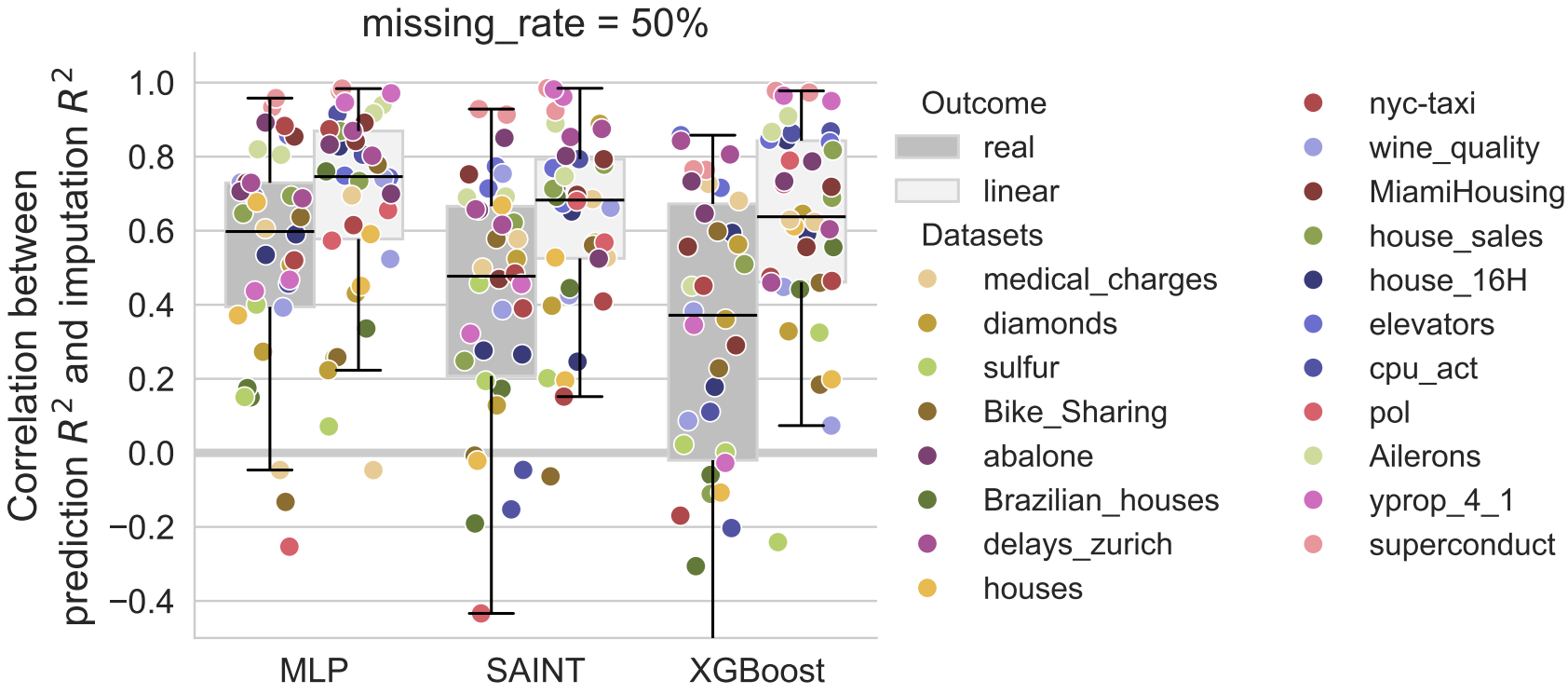


Partial correlation between imputation quality and performance



☞ Good imputations matter less when the response is non-linear.

Partial correlation between imputation quality and performance



☞ Good imputations matter less when the response is non-linear.

☞ Conclusion: Better imputations are not always worth it!

Adding the mask is beneficial, even with MCAR data.

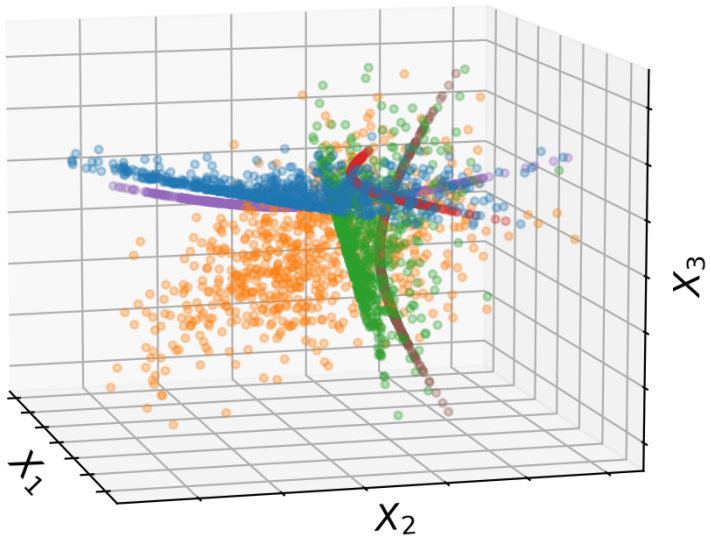
- **This finding has not been reported before.**

- **Current state-of-the-art related to the mask:**
  - In MNAR, adding M is beneficial as it contains relevant information for predicting Y.
  - In MCAR, adding M does not degrade performances asymptotically in linear models (theoretical result).

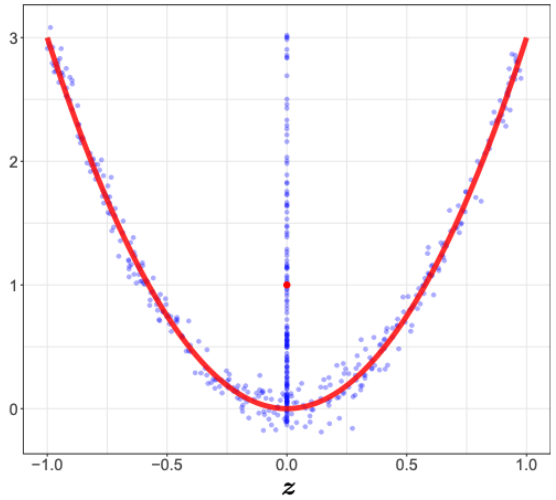
- **It can be surprising at first sight,**
  - as there is absolutely no relevant information in M for predicting Y in MCAR.

Why is M useful in MCAR data?

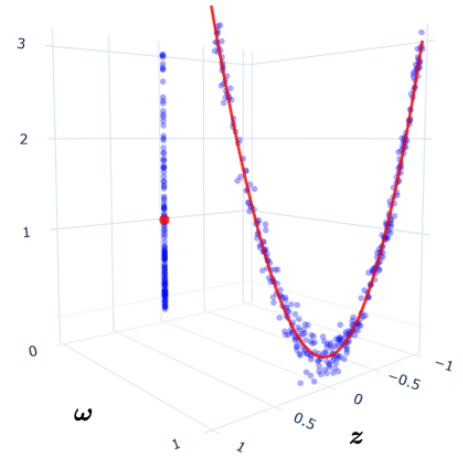
We hypothesize that it simplifies the modeling of functions that exhibit discontinuities at points where the original data patterns of the observations change.



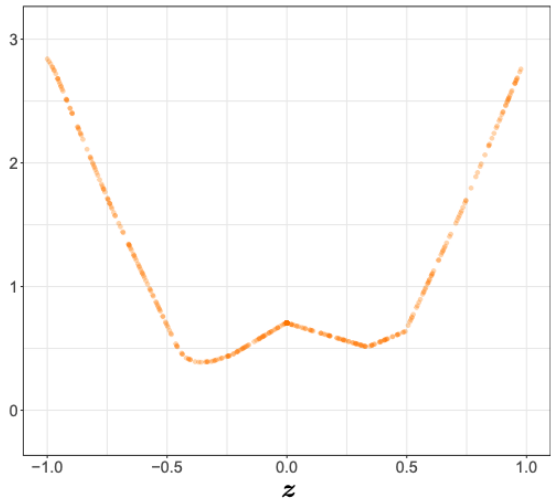
Why is the mask beneficial, even with MCAR data?



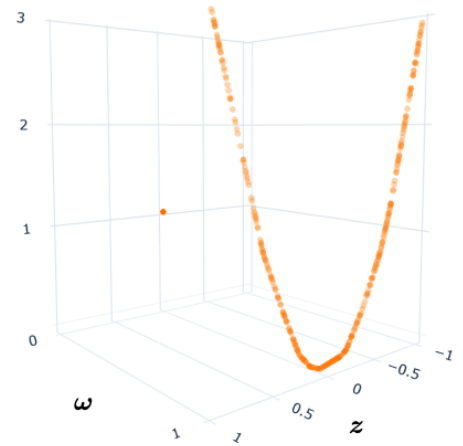
(a) Test data (blue) and Bayes regression function $z \mapsto \mathbb{E}(Y_1 \mid \boldsymbol{Z}_1 = z)$ (red).



(b) Test data (blue) and Bayes regression function $(z, \boldsymbol{\omega}) \mapsto \mathbb{E}(Y_1 \mid \boldsymbol{Z}_1 = z, \boldsymbol{\Omega}_1 = \boldsymbol{\omega})$ (red).



(c) Output of the neural network trained without the revelation vectors.



(d) Output of the neural network trained with the revelation vectors.

from Ma et al, Deep learning with missing data (2025)

Why is the mask beneficial, even with MCAR data?

Missingness indicators are shuffled per sample.

$\Rightarrow$ The number of missing values per sample is preserved.

☞ Using a shuffled missingness indicator harms prediction performance, except for XGBoost for which performances are unchanged.

Benchmark 2:

Identifying the best approaches on real health databases.

Benchmarking missing-values approaches for predictive models on health databases, Gigasciences 2022.

13 tasks across 4 health databases

traumabase.eu

- $n \approx 10,000$
- 5 prediction tasks:
  - death (C)
  - hemorrhagic shock x2 (C)
  - platelet level (R)
  - septic shock (C)

biobank uk
Improving the health of future generations

- $n \approx 300,000$
- 5 prediction tasks:
  - malignant breast neoplasm x2 (C)
  - fluid intelligence score (R)
  - Parkinson (C)
  - Melanoma (C)

MIMIC

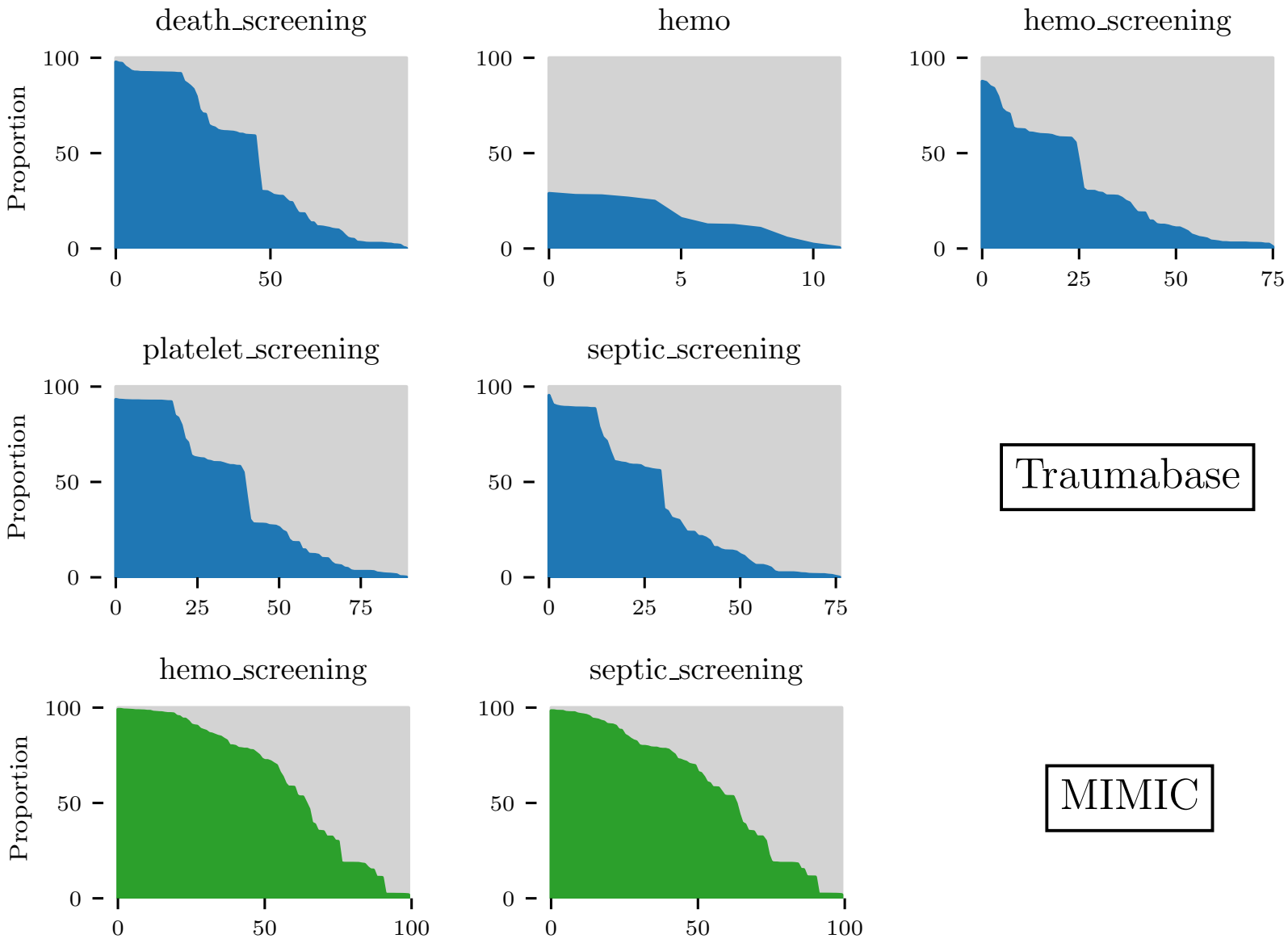
- $n \approx 30,000$
- 2 prediction tasks:
  - hemorrhagic shock (C)
  - septic shock (C)

NHIS
NATIONAL HEALTH INSURANCE SCHEME

- $n \approx 20,000$
- 1 prediction task:
  - yearly income (R)

For all databases, the number of features considered is between 12 and 100.

Missingness levels on MIMIC and the Traumabase



Proportion of missing values for each feature.

Baselines benchmarked

Predictive models

- Gradient Boosted Regression Trees
- Linear models (not shown)

Missing values handling

- Constant imputation
  - mean
  - median
- Conditional imputation
  - Iterative Imputer (MICE like)
  - K-Neareast Neighbours (kNN)
- MIA: Missing Incorporated in Attribute
  - only works with trees
  - does an implicit imputation

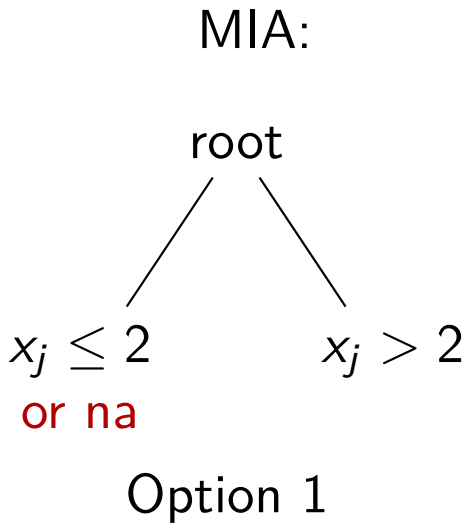
Baselines benchmarked

Predictive models

- Gradient Boosted Regression Trees
- Linear models (not shown)

Missing values handling

- Constant imputation
  - mean
  - median
- Conditional imputation
  - Iterative Imputer (MICE like)
  - K-Neareast Neighbours (kNN)
- MIA: Missing Incorporated in Attribute
  - only works with trees
  - does an implicit imputation



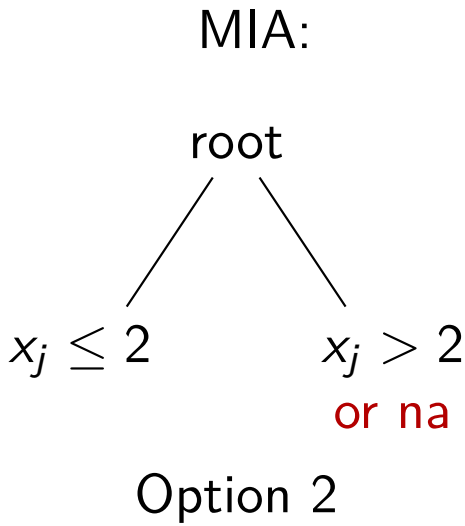
Baselines benchmarked

Predictive models

- Gradient Boosted Regression Trees
- Linear models (not shown)

Missing values handling

- Constant imputation
  - mean
  - median
- Conditional imputation
  - Iterative Imputer (MICE like)
  - K-Neareast Neighbours (kNN)
- MIA: Missing Incorporated in Attribute
  - only works with trees
  - does an implicit imputation



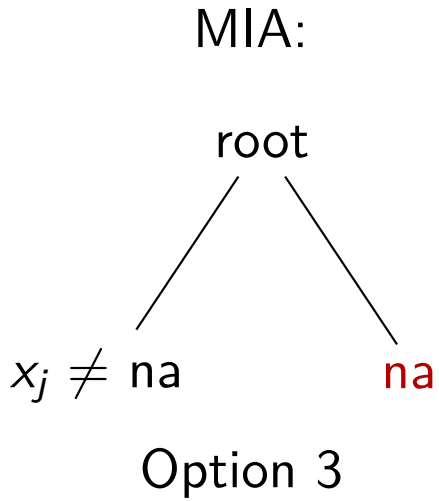
Baselines benchmarked

Predictive models

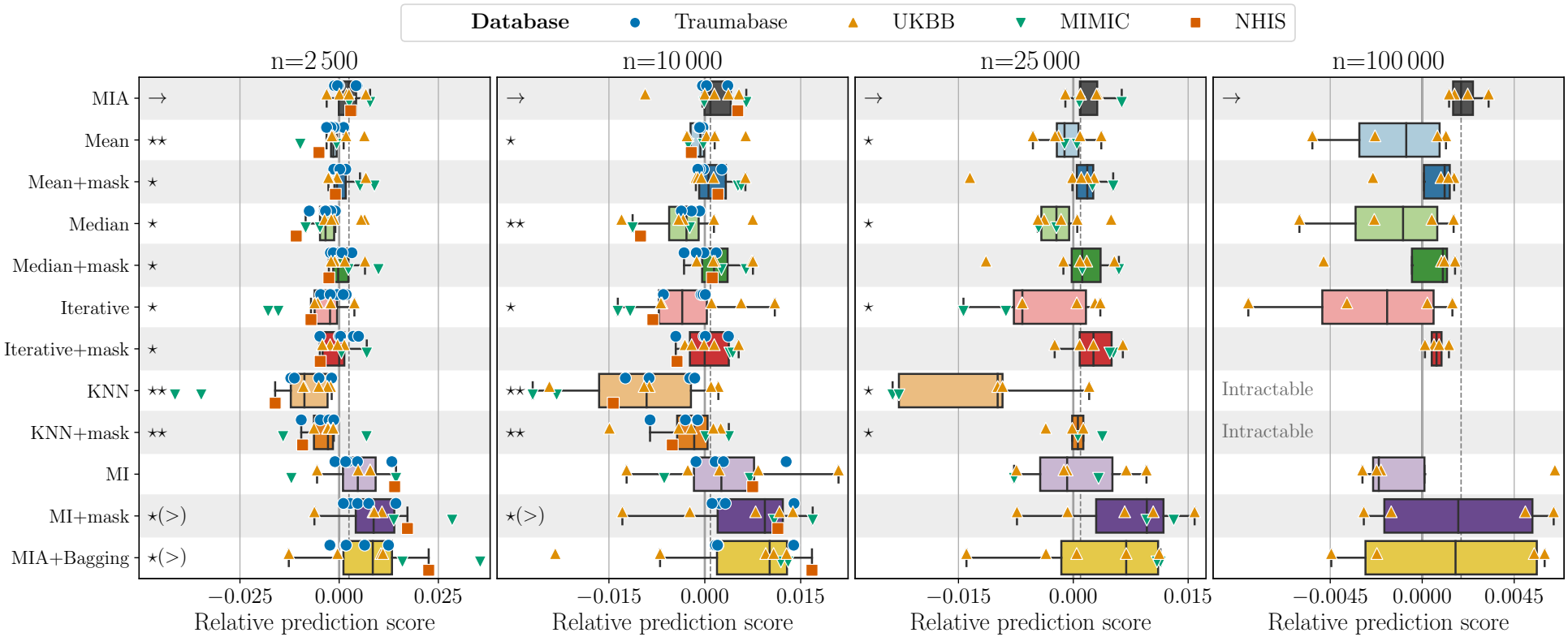
- Gradient Boosted Regression Trees
- Linear models (not shown)

Missing values handling

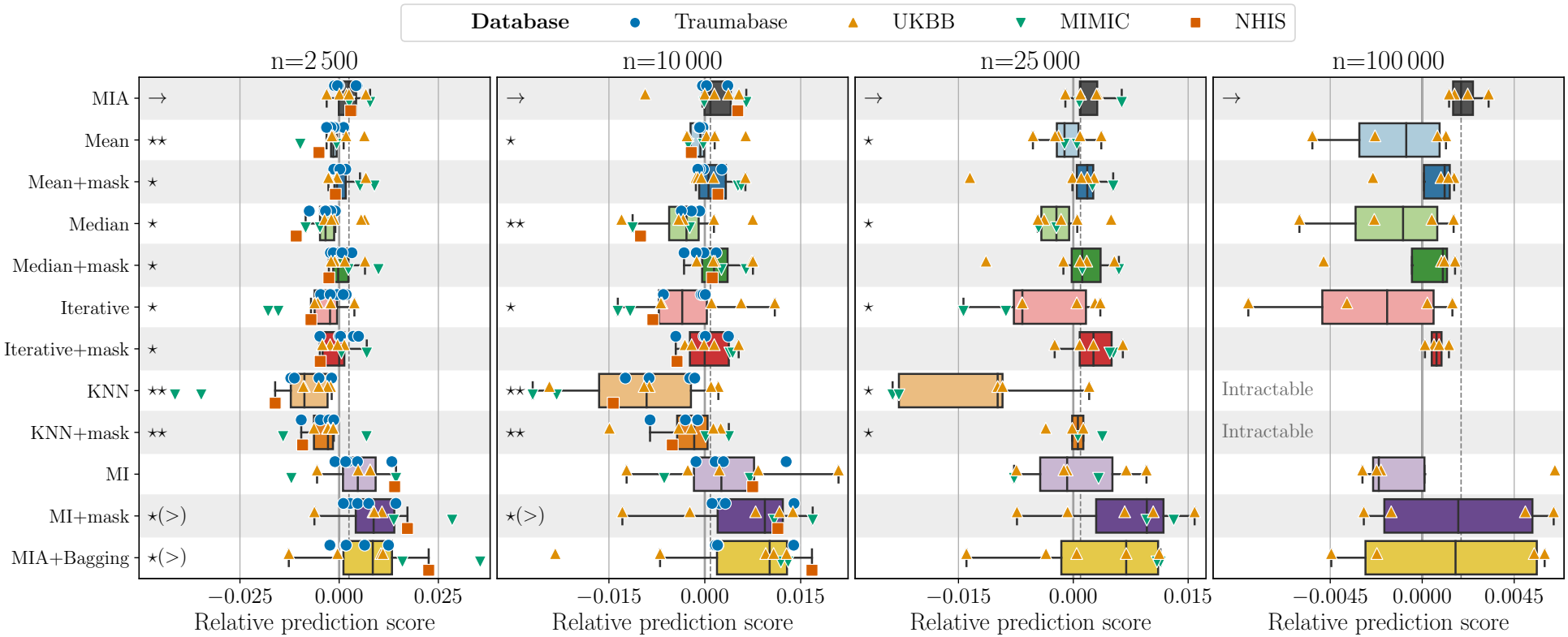
- Constant imputation
  - mean
  - median
- Conditional imputation
  - Iterative Imputer (MICE like)
  - K-Neareast Neighbours (kNN)
- MIA: Missing Incorporated in Attribute
  - only works with trees
  - does an implicit imputation



Prediction performances

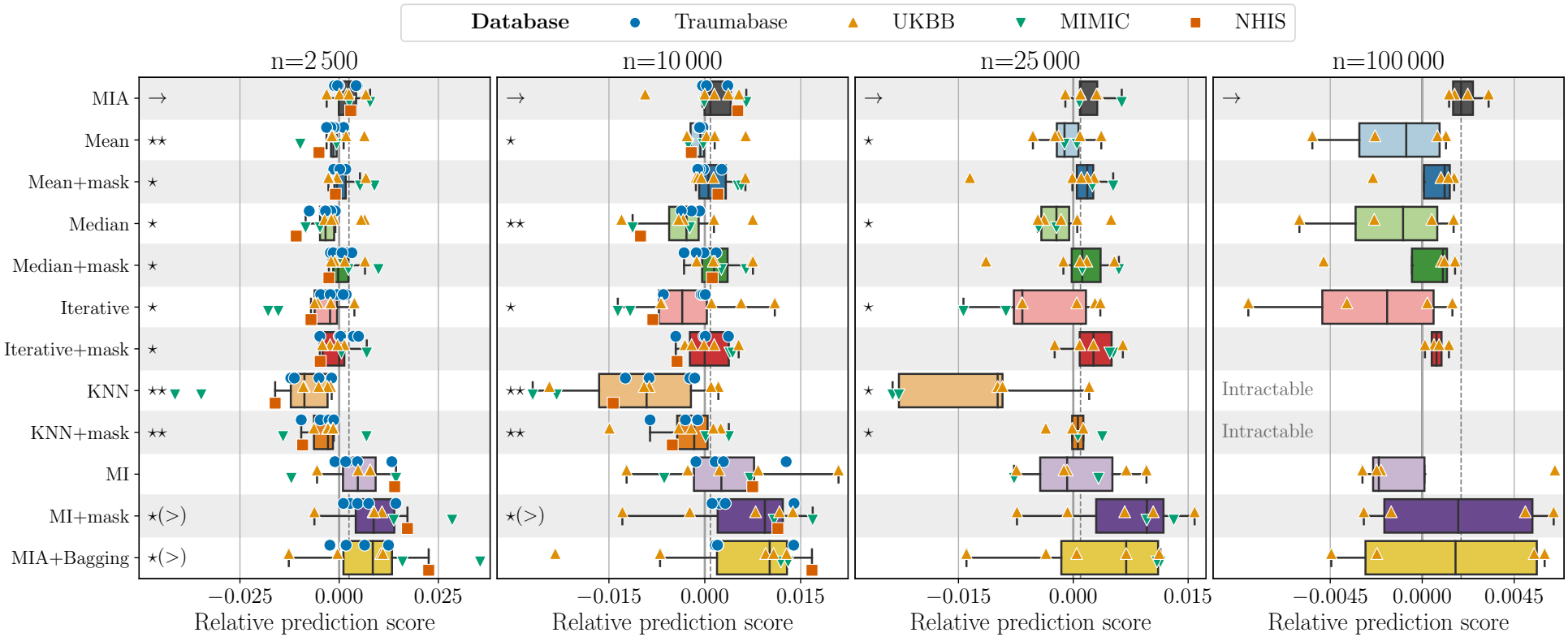


Prediction performances



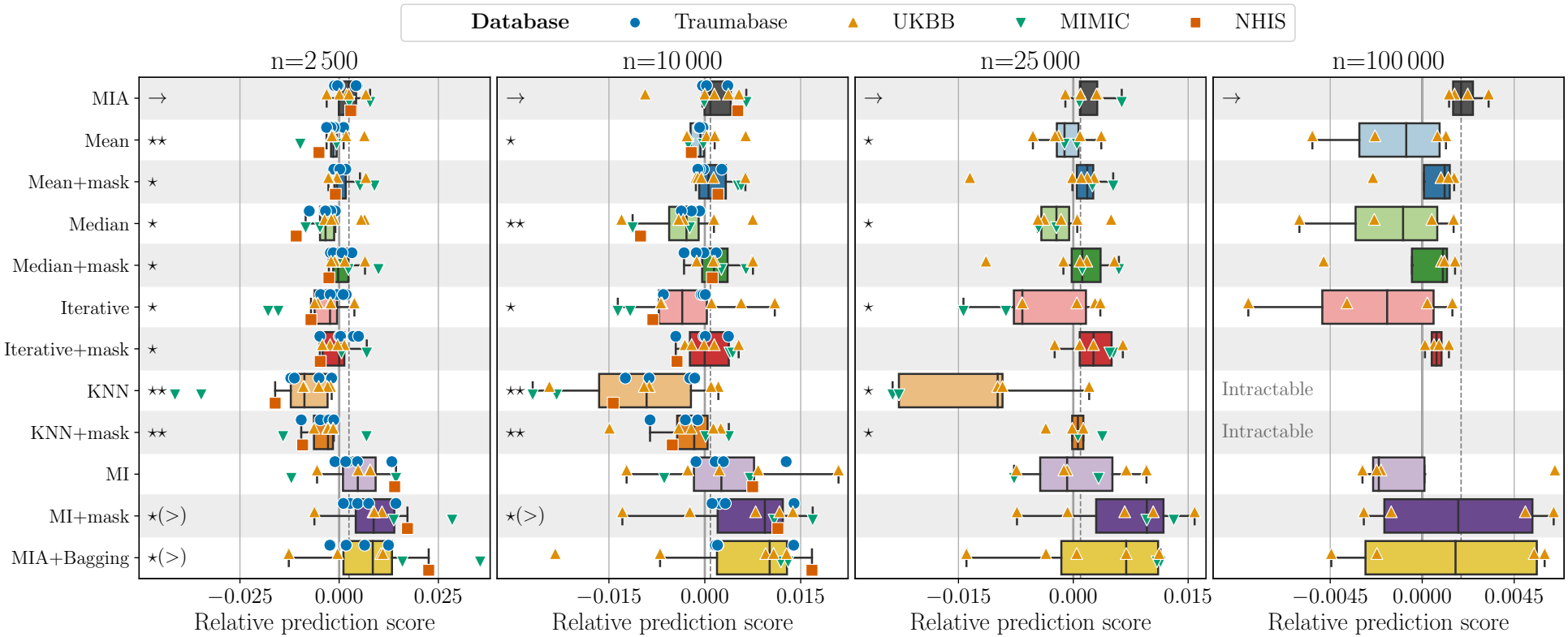
☞ Adding the mask improves prediction ($\Longrightarrow$ informative missingness).

Prediction performances



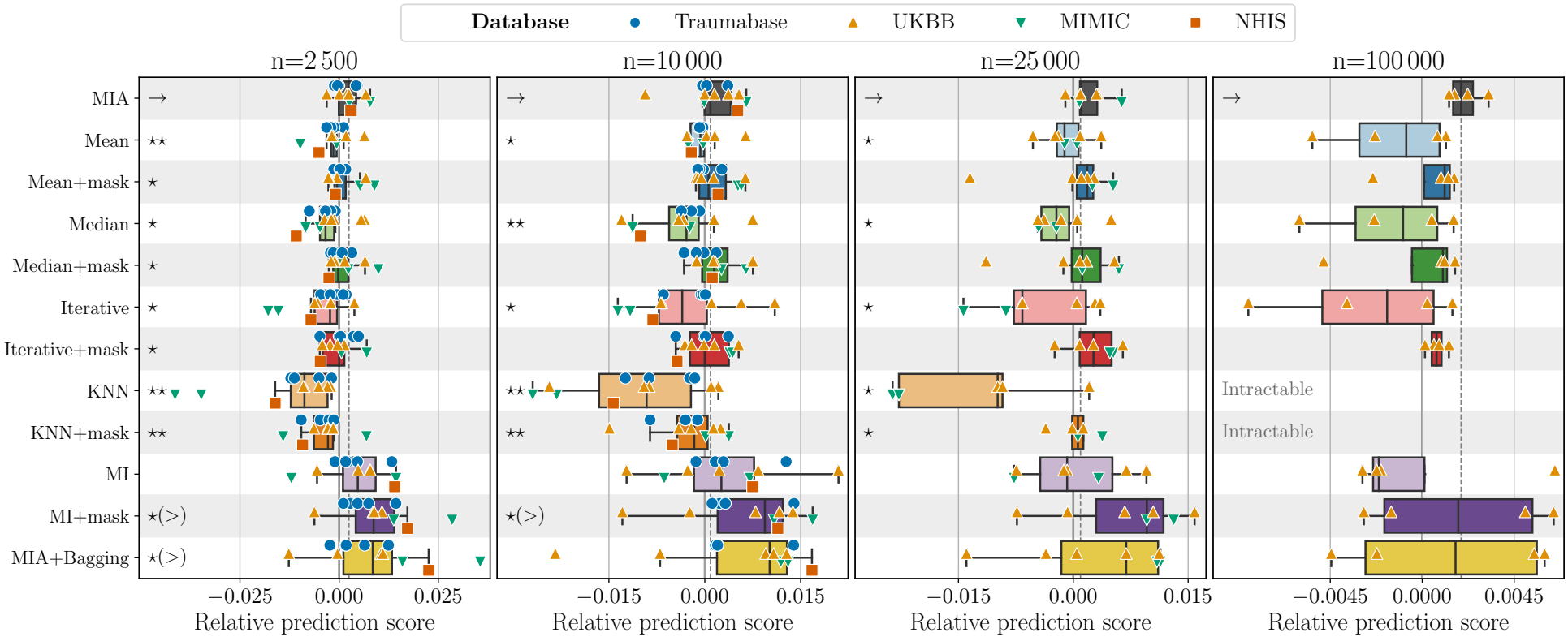
- ☞ Adding the mask improves prediction ($\Longrightarrow$ informative missingness).
- ☞ Conditional imputation is on par with constant imputation

Prediction performances



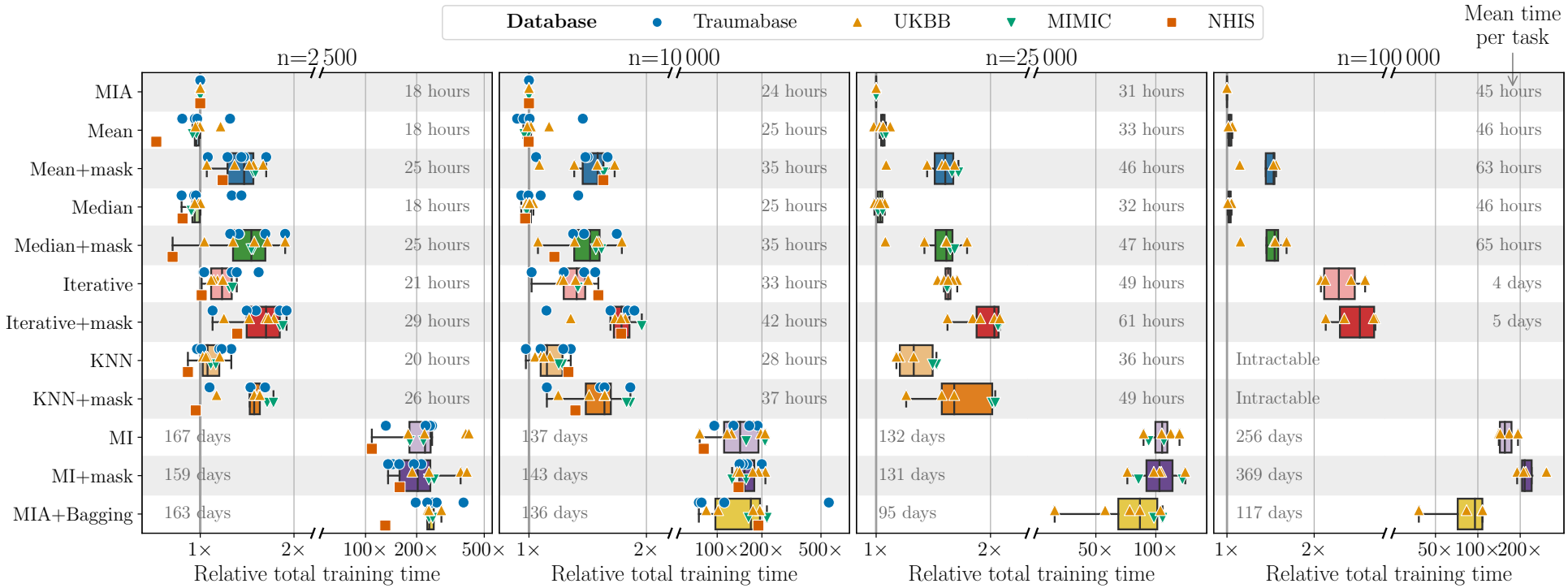
- ☞ Adding the mask improves prediction ($\Longrightarrow$ informative missingness).
- ☞ Conditional imputation is on par with constant imputation
- ☞ Native missing-values support with MIA gives better predictions than imputation.

Prediction performances



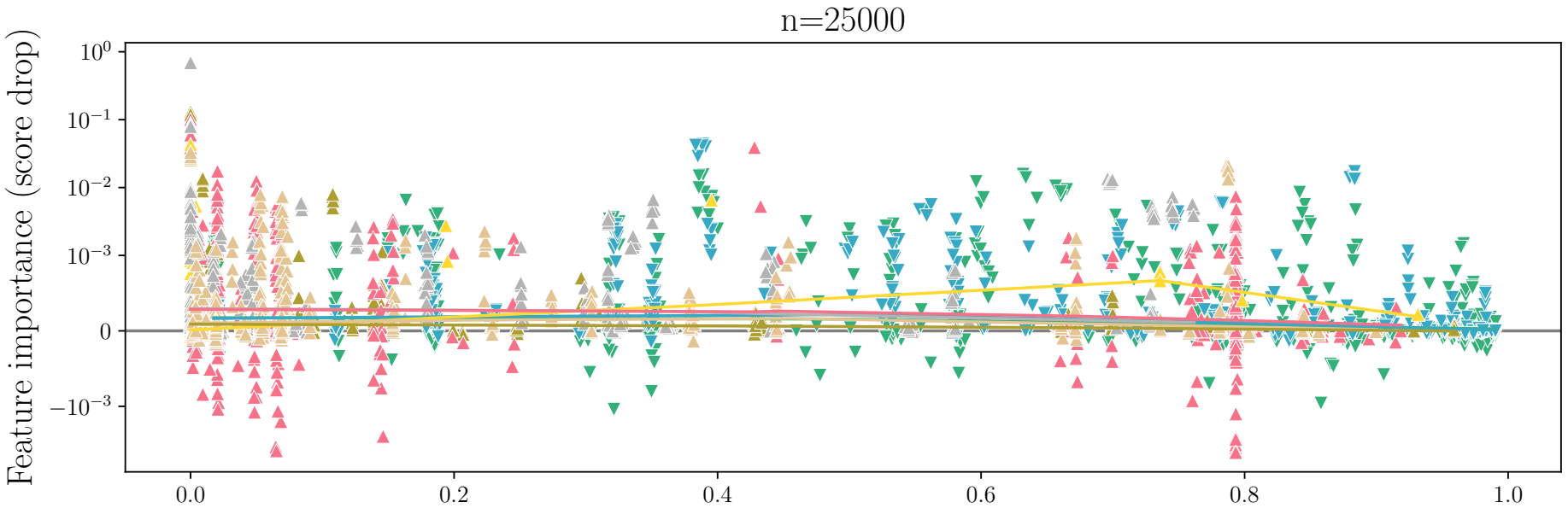
- ☞ Adding the mask improves prediction ($\Longrightarrow$ informative missingness).
- ☞ Conditional imputation is on par with constant imputation
- ☞ Native missing-values support with MIA gives better predictions than imputation.
- ☞ Bagging improves prediction performance but with a prohibitive time cost.

Prediction performances



- ☞ Adding the mask improves prediction ($\Longrightarrow$ informative missingness).
- ☞ Conditional imputation is on par with constant imputation
- ☞ Native missing-values support with MIA gives better predictions than imputation.
- ☞ Bagging improves prediction performance but with a prohibitive time cost.

Features with high missing rates are also important



Feature importances as a function of missingness proportion.

Summary and takeaways

- Impute-then-Regress (or classify) widely used in practice.
- The imputation algorithm should be learned on the train set only.
- MICE (and in particular missforest) are state-of-the-art imputers.
- Better imputations matter, but often marginally.
- Use the missingness indicator as additional input features (option `add_indicator` in scikit-learn's imputers).
- Tree-based models with MIA often provide very good performances.

Thanks for you attention.

Any questions?