Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

When to Impute? Imputation before and during cross-validation

Byron C. Jaeger*[1] | Nicholas J. Tierney[2] | Noah R. Simon[3]

[1]Department of Biostatistics, University of Alabama at Birmingham, Birmingham, Alabama
[2]Very Important Stuff Committee, Institute B, City B, Country B- HELLO
[3]Department of Biostatistics, University of Washington, Seattle, Washington

Correspondence
*Byron C. Jaeger. Email: bcjaeger@uab.edu

Present Address
327M Ryals Public Health Building 1665
University Blvd Birmingham, Alabama
35294-0022

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut elit odio. Donec fermentum tellus neque, vitae fringilla orci pretium vitae. Fusce maximus finibus facilisis. Donec ut ullamcorper turpis. Donec ut porta ipsum. Nullam cursus mauris a sapien ornare pulvinar. Aenean malesuada molestie erat quis mattis. Praesent scelerisque posuere faucibus. Praesent nunc nulla, ullamcorper ut ullamcorper sed, molestie ut est. Donec consequat libero nisi, non semper velit vulputate et. Quisque eleifend tincidunt ligula, bibendum finibus massa cursus eget. Curabitur aliquet vehicula quam non pulvinar. Aliquam facilisis tortor nec purus finibus, sit amet elementum eros sodales. Ut porta porttitor vestibulum. Integer molestie, leo ut maximus aliquam, velit dui iaculis nibh, eget hendrerit purus risus sit amet dolor. Sed sed tincidunt ex. Curabitur imperdiet egestas tellus in iaculis. Maecenas ante neque, pretium vel nisl at, lobortis lacinia neque. In gravida elit vel volutpat imperdiet. Sed ut nulla arcu. Proin blandit interdum ex sit amet laoreet. Phasellus efficitur, sem hendrerit mattis dapibus, nunc tellus ornare nisi, nec eleifend enim nibh ac ipsum. Aenean tincidunt nisl sit amet facilisis faucibus. Donec odio erat, bibendum eu imperdiet sed, gravida luctus turpis.

KEYWORDS:
Class file; LaTeX; Statist. Med.; Rmarkdown;

1 | INTRODUCTION

In evaluating the performance of predictive modeling procedures, it is understood that so-called training error (the predictive error measured on observations used to fit the model) is a poor proxy for generalization error (the performance of the model on future, as-yet-unseen, observations). The training error of a model will often be overly optimistic for the generalization error. For more complex modeling procedures (or on high dimensional data) this over-optimism can be substantial. As such, it is standard practice to use sample-splitting methods to estimate generalization error. These methods train and evaluate models using separate datasets. Cross-validation (CV) is a common sample-splitting technique that partitions a dataset into v folds. In each replicate of CV, a single fold is designated as validation data and the remaining $v - 1$ folds are combined to form a training dataset. After each fold has been used as validation data, results from the v replicates are aggregated to estimate the modeling procedure's generalization error.

Modeling procedures often involve a sequence of steps. For example, data pre-processing steps such as centering and scaling predictor values or filtering out redundant correlated predictor variables may occur before model fitting. To estimate the generalization error of multi-step modeling procedures, it is recommended that the entire sequence of steps be carried out during each replicate of CV to mimic the application of the entire procedure to an independent test set. This is particularly important for supervised steps (*i.e.,* steps that use the outcome variable). It has been stated that unsupervised variable selection steps (*i.e.,* steps that ignore the outcome variable) can be applied before conducting CV without incurring bias.[1] Since this variable selection does not involve the outcome, it does not give predictors an unfair advantage during CV. However, it is unclear whether this computational shortcut generalizes to unsupervised operations that modify values in the training data rather than filter variables out of the training data.

Missing data (MD) occur frequently in applied settings, and several machine learning algorithms (e.g., regression) are incompatible with MD. Imputation is a technique that replaces MD with estimated values. Imputation of MD is often one of the most computationally expensive steps in modeling procedures. For example, the `missForest` imputation algorithm may fit one random forest model for each column that contains MD. Computational expense of applying `missForest` or other imputation strategies with similar complexity during each replicate of CV scales poorly to larger datasets and to higher numbers of folds. High computational cost may lead analysts to prefer more convenient but less effective strategies to handle MD. A more computationally efficient approach would be to implement 'unsupervised imputation' (*i.e.,* imputing MD without accessing outcome information) *before* conducting CV. However, there is a lack of evidence substantiating the claim that unsupervised imputation conducted before CV will provide reliable estimates of a multi-step modeling procedure's generalization error. More importantly, there is a lack of evidence outlining scenarios when unsupervised imputation before CV can lead to biased estimates of model error.

In this manuscript, we develop empirical evidence assessing whether unsupervised imputation instigated before conducting CV (a strategy we will refer to as `I→CV`) can yield reliable model error estimates. We compare estimated model error according to `I→CV` with estimated model error when imputation is applied *during each replicate* of CV (a strategy we will refer to as `CV ↻ I`). To objectively compare the accuracy of model error estimates using `I→CV` and `CV ↻ I`, we compute 'true' model error using an external validation set that contains MD. Both simulated and real data are leveraged to draw these comparisons, and all scripts to generate results have been made publically available on the first author's GitHub. Our analysis also introduces and applies the `ipa` R package (**i**mputation for **p**redictive **a**nalytics), which provides functions to create single or multiple imputed training and testing sets for prediction modeling.

The rest of this manuscript is organized as follows. In Section 5, we summarize results from experiments using real data from public repositories. Results are summarized alongside our recommendations for applied practice in Section 6

2 | MISSING DATA AND CROSS-VALIDATION

MD mechanisms were first formalized by Rubin, who developed a framework to analyze MD that supposes each data point has some probability of being missing. If the probability of missingness is unrelated to the data (*i.e.,* all data are equally likely to be missing), then the data are missing completely at random (MCAR). When the probability of missingness is related to observed variables in the data (*i.e.,* all data within observed groups are equally likely to be missing), the data are missing at random (MAR). If the probability of missingness is determined by reasons that are unknown or unobserved, the data are missing not at random (MNAR). To illustrate, if a doctor did not run labs for a patient because the clinic was too crowded at the time, the patient's data are MCAR. If instead the doctor chose not to measure the patient's labs because the patient was too young, the patient's data are MAR. If the patient missed the appointment because the patient was too sick, the patient's data are MNAR. In the context of statistical learning, previous findings have shown that when data are MNAR, imputation alone is often less efficient than incorporating features that characterize missing patterns (*e.g.,* missingness incorporated as an attribute)[2,3,4]. Since the primary aim of the current study is to assess the differences between two implementations of imputation (*i.e.,* I→ CV and CV ↻ I), we focus analyses on cases where data are MAR or MCAR.

The mechanisms of MD and methods to engage with MD have been studied thoroughly in the context of statistical inference. The primary focus in this setting is to obtain valid test statistics for statistical hypotheses in the presence of MD. Imputation to the mean and, more broadly, MD strategies that create a single imputed value, have been shown to increase type I errors for inferential statistics by artificially reducing the variance of observed data and ignoring the uncertainty attributed to MD, respectively. Multiple imputation, a widely recommended strategy to handle MD for statistical inference, is capable of producing valid test statistics when data are MCAR or MAR because it can simultaneously address the two shortcomings listed in the previous sentence. It is notable that the 'accuracy' of imputed values is not critical for the success of multiple imputation, given sufficient estimates of conditional distributions[5]. Instead, it is the consistency of the estimated covariance matrix for regression coefficients that makes this strategy ideal for statistical inference.

In the context of supervised statistical learning, data often include a training set and an external validation set. commonly ordered steps for model development are (1) data pre-processing, (2) tuning model hyper-parameters, (3) training the model(s) on the full training set, and (4) validating the model(s) using the validation set. Handling MD is a pre-processing task conducted in the training and validation sets, separately. In contrast to statistical inference, single imputation is often used in supervised learning workflows and strategies with greater imputation 'accuracy' are preferred. When k-fold CV is applied, the training data

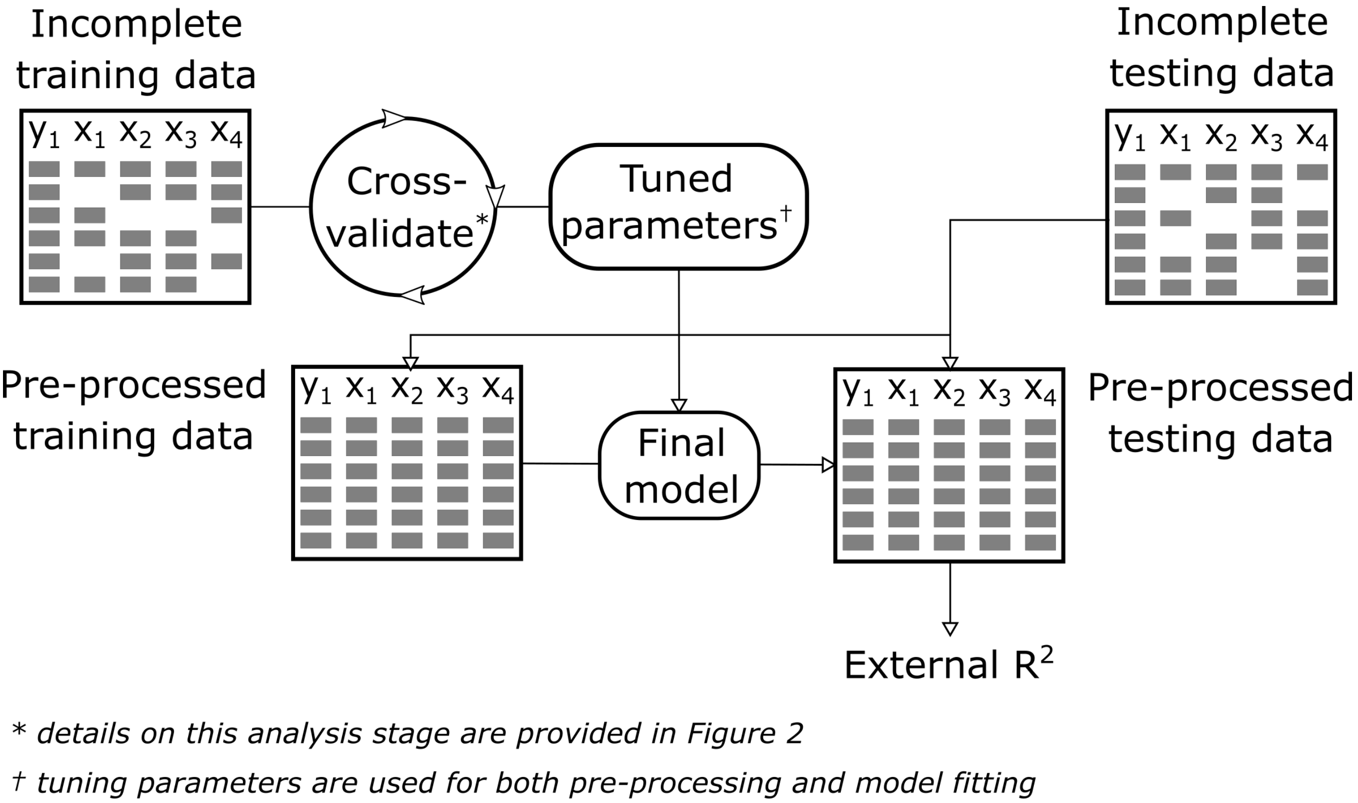
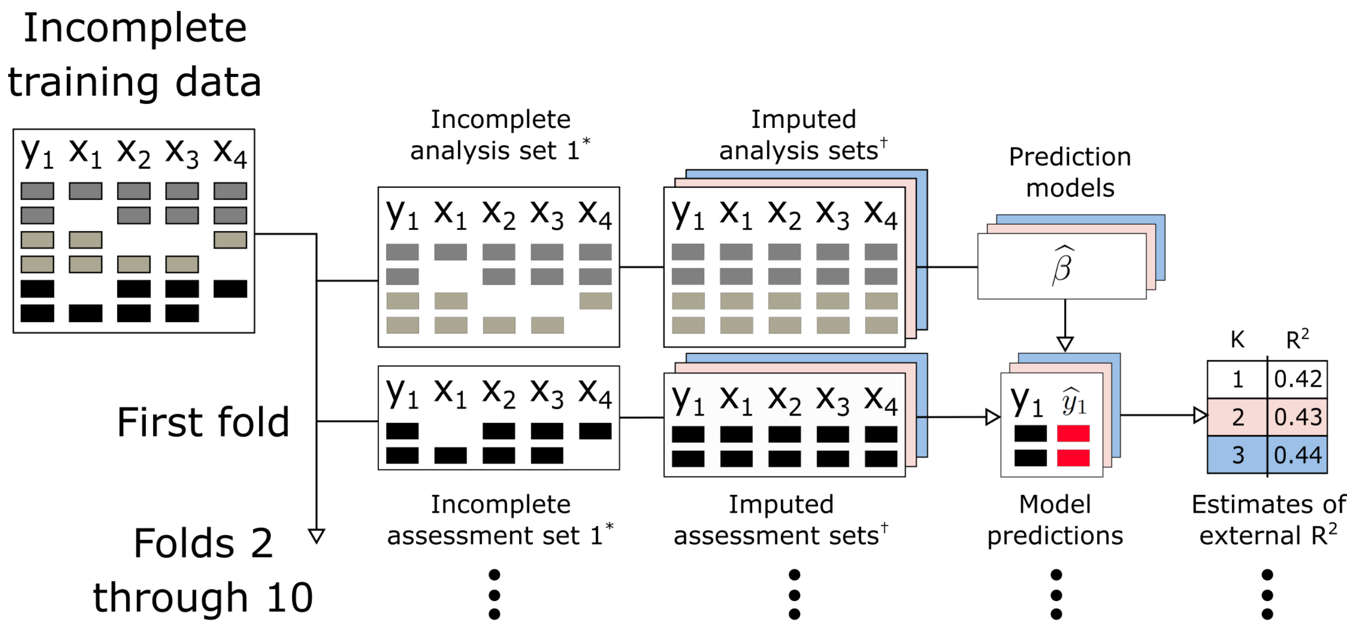

Incomplete training data
y1 x1 x2 x3 x4
Cross-validate*
Tuned parameters†
Incomplete testing data
y1 x1 x2 x3 x4
Pre-processed training data
y1 x1 x2 x3 x4
Final model
y1 x1 x2 x3 x4
Pre-processed testing data
External R2
* details on this analysis stage are provided in Figure 2
† tuning parameters are used for both pre-processing and model fitting

FIGURE 1 A standard workflow for machine learning analysis.

are split into k non-overlapping subsets (*i.e.,* folds). Each fold is then used as a testing set for the model(s) developed on the $k-1$ other folds. Aggregating model errors over all k steps provides an estimate of the modeling procedure's generalization error, making k-fold CV an ideal resampling technique for tuning model hyper-parameters or comparing different modeling procedures. To mimic the application of the entire modeling procedure to an independent validation set, it is recommended that all data pre-processing steps be executed in each replicate of CV. Following CV, the generalization error of the modeling procedure(s) is measured by applying the 'tuned' modeling procedure to the entire training set and then evaluating the resulting prediction function's accuracy in the external validation set.

There are at least two ways to evaluate the effectiveness of an imputation strategy. One approach is to evaluate how accurately the strategy imputes unobserved data. However, imputation accuracy can only be measured when the values of unobserved data are known, which limits its practical use. Another approach is to evaluate the generalization error of models fitted to the imputed data. This approach is applicable in any setting involving supervised learning. Evaluating imputation strategies in this manner also allows imputation parameters to be 'tuned' in the same manner as parameters in the prediction model. The current manuscript focusses exclusively on generalization error. <something about picking the right tuning parameters, and something else about estimating the error with as little bias as possible>



* Analysis and assessment sets play the roles of training and testing, respectively, during cross-validation

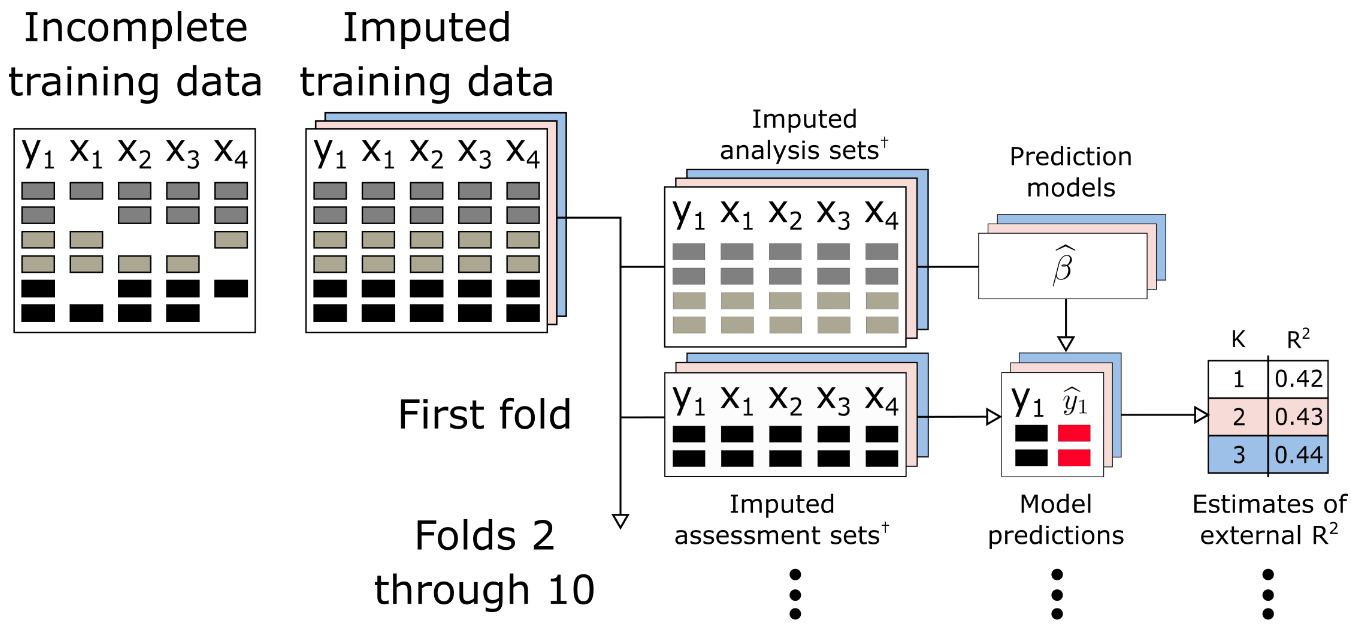
† Imputed datasets are created using 1, 2, and 3 nearest neighbors

FIGURE 2 A traditional workflow for cross-validation that applies imputation within each replicate (*i.e.*, `CV ↻ I`).

3 | ORDER OF OPERATIONS

When employing supervised learning methods with split-sample-validation or cross-validation, it is critical that training data is separated from validation data before any "learning" is done: *The entire supervised pipeline must be run using only training data*. This applies both to the fitting procedure (eg. applying linear regression, or random forests, etc…), and to any supervised preprocessing steps. There are a number of examples in the literature where wildly optimistic estimates of validation error have been obtained because supervised variable selection was performed on the entire dataset, rather than just the training sample (CITE). In scenarios with a larger number of features, even simple methodologies, eg. selecting those features with high individual correlation with outcome, can induce substantial bias (CITE).

It is not unusual to separate training and validation data before performing even unsupervised preprocessing steps. For example, in many supervised learning frameworks, it is common to center and scale features, such that they are mean 0, variance 1 before engaging with the outcome. As this does not involve the outcome, it is entirely unsupervised. Nonetheless to most accurately replicate the "fitting process" in each replicate of CV, it is common to learn the centering and scaling parameters for each feature in each training subset, and apply those parameters to center/scale features in the corresponding validation subset (CITE).



* Analysis and assessment sets play the roles of training and testing, respectively, during cross-validation

† Imputed datasets are created using 1, 2, and 3 nearest neighbors

FIGURE 3 An experimental workflow for cross-validation that applies unsupervised imputation prior to data splitting (*i.e.*, I→CV).

3.1 | Testing data

4 | SIMULATED EXPERIMENTS

The goal of the current simulation study was to assess empirical differences between CV ↻ I and I→CV. Our primary objective was to measure and compare how well each strategy approximated a model's true generalization error. We assessed estimation of true external R^2 using bias, variance, and root-mean-squared error (RMSE). The RMSE provides an overall assessment of estimation accuracy that depends on both bias and variance. A secondary objective was to assess the performance of downstream modeling strategies whose tuning parameters were selected using CV ↻ I and I→CV.

4.1 | Data-generating mechanisms

Consider the linear regression model, where a continuous outcome vector $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$ is generated by a linear combination of predictor variables $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_p\right]$. This functional relationship is often expressed as

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon,$$

where β is a $p \times 1$ vector of regression coefficients and ε is a normally distributed $N \times 1$ zero-mean random error vector. In practice, $\mathbf{X}$ often has some 'junk' variables that are not related to the outcome. We fixed the number of true predictor variables at 10, the standard error of ε at 1, and set $\beta = [-1.00, -0.78, -0.56, -0.33, -0.11, 0.11, 0.33, 0.56, 0.78, 1.00]$ throughout the simulation study. Columns of $\mathbf{X}$ were generated from a multivariate normal distribution with a first order autoregressive correlation structure. Specifically, the correlation between columns $\mathbf{x}_i$ and $\mathbf{x}_j$ was $\rho^{|i-j|}$, where ρ was set to 3/4 throughout the study. We applied this design to generate a training set along with an external validation set comprising 10,000 observations in each simulated replicate.

We created three data-generation 'scenarios'. In scenario 1, the observed data are independent and identically distributed (iid). In scenario 2, the data are iid conditional on an observed grouping variable. A total of 11 groups are formed, one in the validation set and the remaining 10 in the training set. Each group is characterized by a unique mean value for predictor variables. During CV, the observed groups are separated into ten folds to mimic the prediction of outcomes in a population with different characteristics. Scenario 3 is identical to scenario 2 in all regards except that the grouping variable is latent. Consequently, CV does not break the observed groups into separate folds for scenario 3.

Amputing data

We applied the `ampute` function from the `mice` R package to generate missing values in simulated data. In each replicate, 90% of observations comprised at least one missing value. We designated up to p MD patterns randomly in each simulation replicate, where p is the number of non-outcome columns in the simulated data. A MD pattern indicates which of the p predictor variables are set to missing. For each MD pattern, the number of missing variables was randomly set to an integer ranging from 1 to $p/2$. This procedure usually induced missing values in 30-50% of the data. When data were MAR, we applied the default method for the `ampute` function (`ampute.default.weights`) to induce missingness based on the observed variables. Throughout the experiment, we applied the same missing patterns and MD mechanism in the training set and the external validation set.

Modeling procedure

We applied k-nearest-neighbor imputation to handle MD and least absolute shrinkage and selection operator (LASSO) regression to develop prediction functions throughout the simulated experiments. Nearest neighbors in the training set were used to form imputed values in the training and external validation sets. We created one imputed set for each $k \in \{1, 2, \dots, 35\}$. We selected a value for the regularization parameter $lambda$ in each imputed dataset, separately, using 10-fold CV (*i.e.,* `cv.glmnet`). The λ value selected was the one that minimized cross-validated RMSE.

Analysis plan

We varied the scenario (1, 2, or 3; described in a preceding paragraph), missing mechanism (MCAR or MAR), ratio of predictor variables to junk variables (1:1, 1:4, and 1:49), and the number of training observations (N = 100, 500, 1,000, 5,000). We present results for each of 72 settings determined by these parameters and also provide overall summary statistics for scenarios 1, 2, and 3 when data are MCAR and MAR (*i.e.,* aggregating over training sample size and predictor to noise ratio). In each simulation replication, we computed the true external R^2 in the validation set for each potential value of nearest neighbors (*i.e.,* $k \in \{1, 2, \dots, 35\}$). We also estimated external R^2 for each value of k using `CV ↻ I` and `I→ CV`, separately, to evaluate how well these CV procedures estimated the true external R^2. We assessed the difference between estimated external R^2 according to `CV ↻ I` and `I→ CV` as well as the bias, variance, and root-mean-squared error (RMSE) of these estimates. Last, we investigated the accuracy of downstream models when `CV ↻ I` and `I→ CV` were applied to select the number of neighbors to use for imputation and the regularization parameter for a penalized regression model.

4.2 | Results

Overall, a total of 59,966 out of 60,000 (99.94%) simulation replicates were completed over a span of 21,813 computing hours. Incomplete replicates were not analyzed, as these were replicates where at least one of the amputation, imputation, or prediction models did not converge. Across all replicates, the mean number of minutes used to form imputed data using `CV ↻ I` and `I→ CV` were 7.31 and 0.79, respectively, a ratio of 9.24. As a point of reference, using the full training set, the mean number of minutes needed to tune and fit `glmnet` models was 1.31.

Across all scenarios, the mean external R^2 ranged from 0.232 to 0.443 (**Table** 1). External R^2 values were positively correlated with training set size and the ratio of predictor variables to junk variables. Notably, the mean external R^2 values in scenario 1 were uniformly greater than corresponding mean external R^2 values in scenarios 2 and 3, and the maximum difference between mean external R^2 values in scenario 2 versus scenario 3 was 0.001. The mean absolute difference between external R^2 estimates using `CV ↻ I` and `I→ CV` shrunk towards zero as the size of the training set increased (**Table** 2). The differences between `CV ↻ I` and `I→ CV` were lowest in scenario 1 and greatest in scenario 2. These patterns were also present in visual depictions of external R^2 portrayed as a function of k neighbors (**Figure** 4).

Bias, variance, and RMSE

For scenario 1, the overall bias of R^2 estimates under MCAR using `CV ↻ I` was -0.00126 versus 0.00238 using `I→ CV` (**Table** 3). When the data were MAR, the overall biases were 0.00310 for `CV ↻ I` versus -0.00071 for `I→ CV` . In scenarios 2 and 3, the bias of `CV ↻ I` was lower than that of `I→ CV` , and `I→ CV` consistently provided overly optimistics error estimates. The overall standard deviation of R^2 estimates was higher for `CV ↻ I` versus `I→ CV` in all three scenarios and both missing data mechanisms. The difference in standard deviation was most pronounced in scenario 3 when data were MCAR (0.07354 [`CV ↻ I`] versus 0.06784 [`I→ CV`]; **Table** 4). Despite the optimistic bias of `I→ CV` in scenario 2, the reduced variance of this approach lead to a lower overall RMSE for external R^2 compared to `CV ↻ I` (**Table** 5). When the data were MCAR in scenario 2, `CV ↻ I` and `I→ CV` obtained RMSEs of 0.06027 and 0.05890, respectively. Similarly, when the data were MAR in scenario 2, overall RMSE values were 0.06015 and 0.05876.

Downstream model performance

When CV ↻ I and I→CV were applied to select tuning parameters, the overall mean external R^2 was higher using CV ↻ I in 6 out of 6 comparisons (**Table** 6). The greatest overall difference in mean R^2 between downstream models occurred in scenario 3 when the data were MAR (absolute difference in model R^2: 0.00033; relative difference in model R^2 : 0.09%).

5 | REAL DATA EXPERIMENTS

6 | DISCUSSION AND RECOMMENDATIONS

We demonstrated empirical properties of CV ↻ I and I→ CV using nearest-neighbor imputation and LASSO regression. We selected these methods because they have been studied thoroughly and are widely used in applied settings. To generate data that matched the structure of our modeling procedure, we generated outcomes using linear effects without interaction. We studied three broad scenarios that were relevant to CV: Scenario 1 was an ideal setting where I→CV and CV ↻ I should have provided almost identical estimates of generalization error. Scenarios 2 and 3 were meant to test whether I→CV produced biased estimates of generalization error because in settings where I→ CV clearly did not mimic the final application of a trained model to an external validation set. Remarkably, despite its bias in scenario 2, the reduction in variance of R^2 estimates using I→CV lead to a lower overall RMSE compared to CV ↻ I. Downstream model performance was consistently superior when CV ↻ I was used instead of I→CV. However, the increase in performance was modest (maximum overall relative difference in external R^2: 0.09%).

TABLE 1 True external R^2 values for the modeling technique that is internally assessed using `CV` ↻ `I` and `I`→ `CV`.

	Scenario 1		Scenario 2		Scenario 3	
N	MAR	MCAR	MAR	MCAR	MAR	MCAR
10 predictors, 10 junk						
100	37.8 (3.44)	37.7 (3.43)	33.4 (6.43)	33.2 (6.62)	33.5 (6.45)	33.2 (6.60)
500	42.8 (2.93)	42.7 (2.94)	39.8 (5.51)	39.6 (5.56)	39.8 (5.54)	39.6 (5.57)
1000	43.6 (2.93)	43.5 (2.93)	40.9 (4.94)	40.7 (4.92)	40.9 (4.94)	40.7 (4.93)
5000	44.3 (3.00)	44.2 (3.00)	41.9 (4.53)	41.7 (4.65)	41.9 (4.53)	41.7 (4.66)
10 predictors, 40 junk						
100	34.6 (3.79)	34.5 (3.69)	30.3 (7.18)	30.3 (6.95)	30.3 (7.28)	30.3 (7.01)
500	40.6 (2.76)	40.6 (2.77)	38.2 (4.78)	38.1 (4.96)	38.2 (4.80)	38.1 (4.97)
1000	41.5 (2.73)	41.5 (2.74)	39.2 (4.72)	39.1 (4.79)	39.2 (4.71)	39.1 (4.78)
5000	42.6 (2.75)	42.6 (2.74)	40.4 (4.50)	40.3 (4.54)	40.4 (4.50)	40.3 (4.55)
10 predictors, 490 junk						
100	27.7 (5.01)	27.8 (4.94)	23.3 (6.92)	23.2 (6.69)	23.3 (6.73)	23.2 (6.70)
500	37.5 (2.94)	37.5 (2.93)	35.7 (4.30)	35.8 (4.31)	35.8 (4.28)	35.8 (4.31)
1000	38.8 (2.84)	38.8 (2.83)	37.2 (4.55)	37.1 (4.55)	37.1 (4.57)	37.1 (4.56)
5000	39.8 (2.78)	39.8 (2.79)	38.5 (4.41)	38.5 (4.30)	38.5 (4.37)	38.5 (4.30)
Overall						
—	39.3 (5.48)	39.3 (5.43)	36.6 (7.38)	36.5 (7.35)	36.6 (7.36)	36.5 (7.35)

All values are scaled by 100 for convenience

TABLE 2 Mean absolute differences in estimates of external R^2 between `CV` ↻ `I` and `I`→ `CV`.

	Scenario 1		Scenario 2		Scenario 3	
N	MAR	MCAR	MAR	MCAR	MAR	MCAR
10 predictors, 10 junk						
100	1.20 (1.03)	1.19 (0.98)	2.40 (1.73)	2.42 (1.74)	1.44 (1.21)	1.45 (1.15)
1000	0.21 (0.20)	0.21 (0.20)	1.17 (1.09)	1.24 (1.17)	0.22 (0.20)	0.23 (0.20)
500	0.31 (0.29)	0.32 (0.29)	1.31 (1.09)	1.39 (1.18)	0.35 (0.30)	0.35 (0.30)
5000	0.09 (0.08)	0.09 (0.09)	1.01 (0.86)	1.07 (0.97)	0.09 (0.09)	0.09 (0.09)
10 predictors, 40 junk						
100	1.48 (1.26)	1.45 (1.27)	2.64 (1.89)	2.67 (1.92)	1.74 (1.43)	1.79 (1.44)
1000	0.22 (0.20)	0.23 (0.20)	1.23 (1.08)	1.29 (1.16)	0.23 (0.20)	0.23 (0.20)
500	0.34 (0.30)	0.34 (0.29)	1.38 (1.22)	1.42 (1.27)	0.36 (0.30)	0.35 (0.30)
5000	0.09 (0.09)	0.10 (0.09)	1.11 (1.01)	1.20 (1.13)	0.10 (0.09)	0.10 (0.09)
10 predictors, 490 junk						
100	2.12 (1.70)	2.00 (1.63)	3.04 (2.24)	3.02 (2.15)	2.38 (1.82)	2.47 (1.89)
1000	0.21 (0.19)	0.21 (0.19)	1.12 (1.16)	1.12 (1.14)	0.21 (0.18)	0.20 (0.18)
500	0.34 (0.30)	0.34 (0.29)	1.20 (1.08)	1.18 (1.06)	0.33 (0.28)	0.33 (0.29)
5000	0.09 (0.08)	0.09 (0.08)	1.09 (1.06)	1.16 (1.13)	0.09 (0.08)	0.09 (0.08)
Overall						
—	0.56 (0.95)	0.55 (0.91)	1.56 (1.51)	1.60 (1.53)	0.63 (1.07)	0.64 (1.09)

All values are scaled by 100 for convenience

TABLE 3 Bias of external R^2 estimates using `CV ↻ I` and `I→CV`

	Missing completely at random						Missing at random					
	Scenario 1		Scenario 2		Scenario 3		Scenario 1		Scenario 2		Scenario 3	
N	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV
10 predictors, 10 junk												
100	0.28	-0.51	0.89	-1.29	-0.42	-1.49	0.54	-0.28	1.09	-1.06	-0.25	-1.33
500	0.09	-0.02	-0.02	-1.34	-1.28	-1.42	0.14	0.05	0.12	-1.12	-1.09	-1.23
1000	0.08	0.03	0.18	-1.03	-1.04	-1.11	0.08	0.03	0.26	-0.88	-0.90	-0.97
5000	-0.03	-0.03	0.01	-1.06	-1.09	-1.11	-0.02	-0.02	0.14	-0.86	-0.91	-0.92
10 predictors, 40 junk												
100	0.65	-0.47	0.87	-1.59	-0.38	-1.92	0.90	-0.28	1.24	-1.20	-0.12	-1.57
500	0.19	0.04	0.49	-0.85	-0.77	-0.94	0.21	0.07	0.63	-0.68	-0.62	-0.78
1000	0.07	0.00	0.14	-1.11	-1.09	-1.17	0.07	0.00	0.18	-1.02	-1.00	-1.09
5000	0.07	0.04	-0.06	-1.25	-1.24	-1.28	0.06	0.04	-0.02	-1.13	-1.13	-1.17
10 predictors, 490 junk												
100	1.06	-0.74	1.74	-1.15	1.07	-1.26	0.97	-0.97	1.55	-1.36	0.88	-1.34
500	0.22	0.07	0.23	-0.80	-0.70	-0.84	0.36	0.19	0.39	-0.65	-0.56	-0.70
1000	0.17	0.10	0.08	-0.96	-0.95	-1.02	0.36	0.29	0.26	-0.77	-0.78	-0.85
5000	0.00	-0.02	0.18	-0.96	-0.97	-0.99	0.05	0.03	0.17	-0.90	-0.92	-0.93
Overall												
—	0.24	-0.13	0.39	-1.12	-0.74	-1.21	0.31	-0.07	0.50	-0.97	-0.62	-1.07

All values are scaled by 100 for convenience

TABLE 4 Standard deviation of external R^2 estimates using `CV ↻ I` and `I→ CV`.

	Missing completely at random						Missing at random					
	Scenario 1		Scenario 2		Scenario 3		Scenario 1		Scenario 2		Scenario 3	
N	`CV ↻ I`	`I→ CV`	`CV ↻ I`	`I→ CV`	`CV ↻ I`	`I→ CV`	`CV ↻ I`	`I→ CV`	`CV ↻ I`	`I→ CV`	`CV ↻ I`	`I→ CV`
10 predictors, 10 junk												
100	6.45	6.41	7.45	7.05	7.10	6.92	6.38	6.31	7.45	7.14	7.18	6.98
500	3.59	3.63	4.05	3.94	3.87	3.92	3.57	3.62	4.03	3.93	3.87	3.92
1000	3.32	3.36	3.78	3.63	3.57	3.60	3.32	3.35	3.76	3.63	3.58	3.61
5000	3.07	3.09	3.35	3.26	3.24	3.25	3.06	3.08	3.32	3.25	3.23	3.24
10 predictors, 40 junk												
100	6.81	6.66	7.38	6.95	7.07	6.87	6.67	6.57	7.22	6.84	6.97	6.73
500	3.53	3.57	3.79	3.65	3.60	3.65	3.54	3.58	3.73	3.62	3.61	3.65
1000	3.13	3.16	3.25	3.20	3.16	3.20	3.14	3.17	3.21	3.18	3.16	3.19
5000	2.82	2.84	3.01	2.87	2.85	2.87	2.82	2.84	2.97	2.87	2.86	2.88
10 predictors, 490 junk												
100	7.66	7.43	8.07	7.67	7.95	7.61	7.58	7.35	8.09	7.75	7.95	7.62
500	3.67	3.71	3.81	3.77	3.74	3.77	3.69	3.72	3.76	3.73	3.72	3.75
1000	3.25	3.27	3.28	3.22	3.20	3.22	3.24	3.27	3.29	3.24	3.21	3.24
5000	2.86	2.88	2.95	2.87	2.86	2.87	2.86	2.88	2.93	2.88	2.87	2.88
Overall												
—	6.48	6.10	7.39	6.84	7.35	6.78	6.50	6.10	7.38	6.83	7.32	6.77

All values are scaled by 100 for convenience

TABLE 5 Root-mean-squared error of external R^2 estimates using CV ↻ I and I→ CV

	Missing completely at random						Missing at random					
	Scenario 1		Scenario 2		Scenario 3		Scenario 1		Scenario 2		Scenario 3	
N	CV ↻ I	I→ CV	CV ↻ I	I→ CV	CV ↻ I	I→ CV	CV ↻ I	I→ CV	CV ↻ I	I→ CV	CV ↻ I	I→ CV
10 predictors, 10 junk												
100	5.89	5.82	9.12	8.86	8.76	8.75	5.81	5.66	8.95	8.69	8.64	8.62
500	2.20	2.19	5.55	5.51	5.50	5.54	2.16	2.15	5.49	5.43	5.43	5.48
1000	1.65	1.65	4.56	4.41	4.41	4.43	1.66	1.66	4.61	4.45	4.45	4.47
5000	0.83	0.83	3.75	3.73	3.74	3.75	0.82	0.83	3.62	3.57	3.57	3.58
10 predictors, 40 junk												
100	6.46	6.30	9.45	9.19	9.24	9.27	6.46	6.31	9.57	9.30	9.44	9.40
500	2.35	2.34	5.11	4.94	4.91	4.94	2.33	2.32	4.95	4.74	4.75	4.77
1000	1.66	1.65	4.57	4.54	4.51	4.54	1.68	1.68	4.49	4.45	4.43	4.45
5000	0.83	0.83	4.07	4.02	4.03	4.03	0.81	0.81	3.95	3.92	3.92	3.94
10 predictors, 490 junk												
100	7.29	7.22	9.13	8.92	8.99	8.98	7.35	7.32	9.30	9.18	8.89	8.89
500	2.40	2.39	4.41	4.35	4.32	4.35	2.39	2.37	4.30	4.20	4.17	4.19
1000	1.75	1.73	4.24	4.14	4.16	4.17	1.74	1.73	4.20	4.05	4.09	4.11
5000	0.85	0.85	3.69	3.59	3.59	3.60	0.86	0.87	3.81	3.72	3.68	3.68
Overall												
—	3.61	3.56	6.03	5.89	5.89	5.90	3.61	3.56	6.02	5.88	5.84	5.85

All values are scaled by 100 for convenience

TABLE 6 Mean external R^2 when `CV ↻ I` and `I→CV` were applied to tune the number of neighbors used for imputation.

	Missing completely at random						Missing at random					
	Scenario 1		Scenario 2		Scenario 3		Scenario 1		Scenario 2		Scenario 3	
N	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV	CV ↻ I	I→CV
10 predictors, 10 junk												
100	38.2	38.1	33.9	33.9	34.0	34.0	38.3	38.3	34.0	34.1	34.0	34.0
500	43.8	43.8	40.5	40.6	40.6	40.6	43.9	43.9	40.8	40.8	40.8	40.8
1000	44.8	44.8	41.9	41.9	41.9	41.9	44.9	44.9	42.0	42.1	42.1	42.1
5000	45.8	45.8	43.2	43.2	43.2	43.2	45.9	45.9	43.4	43.4	43.4	43.4
10 predictors, 40 junk												
100	35.1	35.0	30.9	30.8	30.9	30.9	35.1	35.0	30.9	30.6	30.9	30.7
500	41.8	41.8	39.1	39.1	39.1	39.1	41.8	41.8	39.2	39.2	39.2	39.2
1000	42.8	42.8	40.2	40.3	40.2	40.3	42.8	42.8	40.3	40.3	40.4	40.3
5000	44.0	44.0	41.6	41.7	41.7	41.7	44.0	44.0	41.7	41.7	41.7	41.7
10 predictors, 490 junk												
100	29.0	28.7	24.1	23.8	24.0	23.7	28.8	28.6	23.9	23.7	24.2	23.9
500	39.1	39.1	37.1	37.2	37.1	37.2	39.1	39.1	37.1	37.2	37.1	37.2
1000	40.4	40.4	38.6	38.6	38.6	38.6	40.4	40.4	38.6	38.6	38.6	38.6
5000	41.5	41.5	40.1	40.1	40.1	40.1	41.5	41.5	40.0	40.0	40.1	40.1
Overall												
—	40.5	40.5	37.6	37.6	37.6	37.6	40.5	40.5	37.7	37.6	37.7	37.7

All values are scaled by 100 for convenience

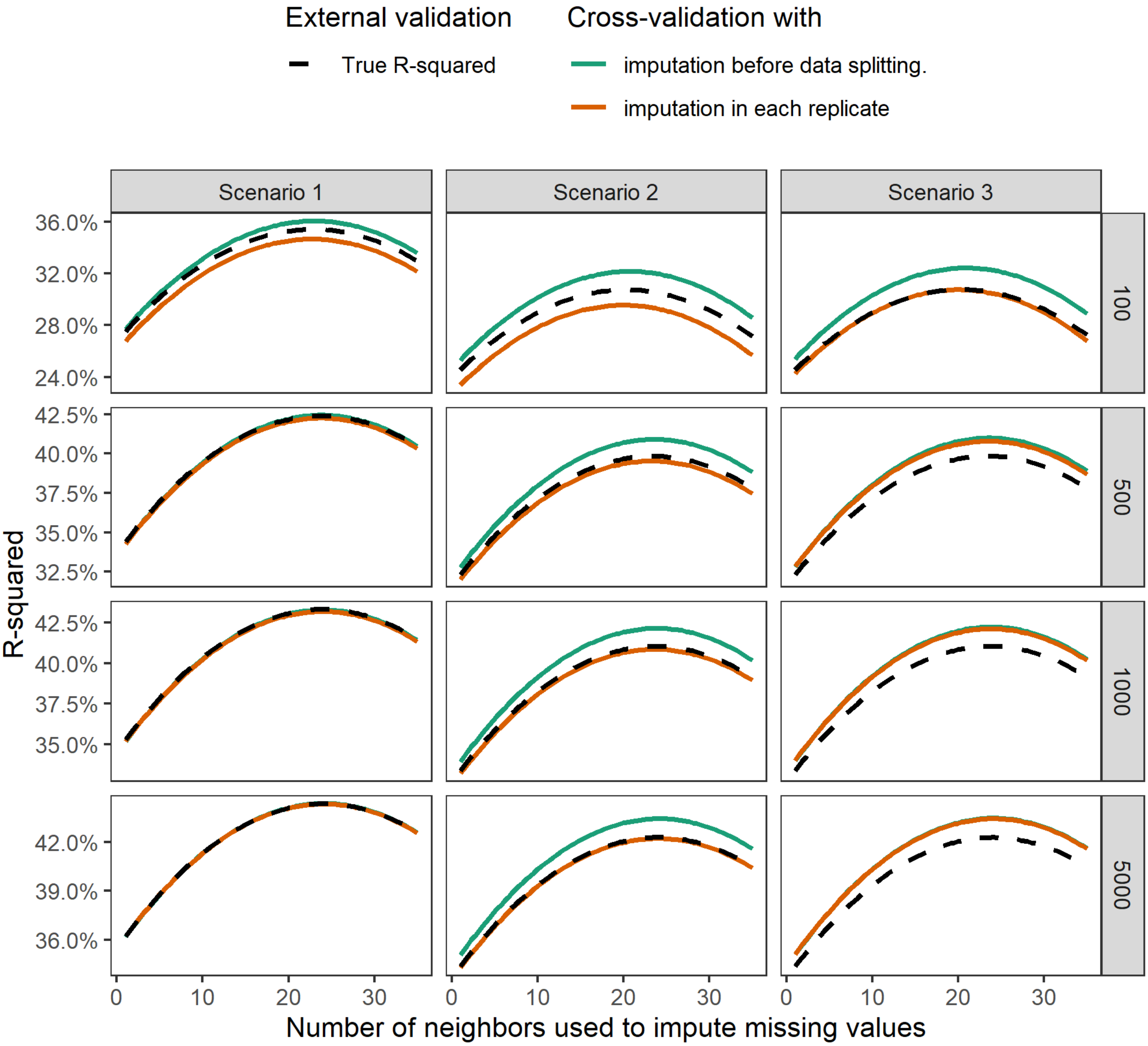


FIGURE 4 External generalization error and internal estimates of generalization error using I→ CV and CV ↻ I

References

1. Hastie T, Tibshirani R, Friedman J. *The elements of statistical learning: data mining, inference, and prediction*. 2. Springer Science & Business Media . 2009.

2. Twala B, Jones M, Hand DJ. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters* 2008; 29(7): 950–956.

3. Twala B. An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence* 2009; 23(5): 373–405.

4. Tang F, Ishwaran H. Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 2017; 10(6): 363–377.

5. Van Buuren S. *Flexible imputation of missing data*. CRC press . 2018.

□