Missingness indicators gain predictive value and lose transferability under class-dependent omission shifts: A Dry Bean classification study

Abstract

Missingness indicators can improve classification by preserving information about which measurements are absent. Their value may change when the relation between missingness and the outcome changes. We conducted a controlled stress test using 13,543 distinct feature vectors from the public Dry Bean dataset, comprising 16 complete numerical measurements and seven classes. We imposed class-dependent omissions while calibrating each feature's expected marginal omission probability to 0.10, 0.30 or 0.50. Logistic regression and ExtraTrees models received median-imputed values, the same values with explicit missingness indicators, or indicators alone. Twelve paired stratified split and mask repetitions compared matched, attenuated, erased and reversed class associations. With omission probability 0.30 and class-dependent training, indicators increased balanced accuracy by 3.14 percentage points for logistic regression and 0.68 for ExtraTrees under matched omissions. Reversing the class association changed these effects to -20.75 and -23.72 points, respectively; the full repetition ranges were [-24.84, -17.53] and [-27.05, -20.36]. Erasing the association also reduced the incremental value of indicators. However, indicators benefited logistic regression trained with outcome-independent omissions, showing that their usefulness was not restricted to learning the artificial class code. Indicators alone exposed the code's strong predictive information and its failure after reversal. Probability losses and class-specific recall revealed consequences hidden by aggregate accuracy. These findings illustrate established missingness-shift risks in a finite multiclass setting: a gain under matched omissions does not establish transferability, and a stable expected marginal missing rate does not ensure stable predictive information. The results concern artificial omissions on one dataset and do not estimate performance in an observed deployment setting.

Introduction

A missing measurement changes both the information available to a classifier and the representation on which it learns. Replacing missing numerical values with a constant permits prediction, while adding a binary indicator for each measurement explicitly represents its absence. The resulting model can use missingness as a predictor. This can be useful when the process that produces absent measurements carries outcome information, but usefulness during model development does not establish usefulness after that process changes. The relevant question is therefore conditional: what does adding explicit missingness information contribute under a specified training mechanism, and how does that contribution transfer to a different evaluation mechanism?

Prediction with incomplete covariates differs from estimation of complete-data parameters. Simple impute-then-learn procedures can have a sound predictive basis under suitable assumptions, even when the imputed values are not adequate for recovering those parameters [1]. In a study of clinical prediction, Sisk and colleagues found conditional benefits from indicators and emphasized consistency between model development and deployment; their experiments held the missingness mechanism stable [2]. Lenz and colleagues reported frequent indicator benefits across naturally incomplete classification datasets, with differences between learners [3]. These results motivate explicit representation, but do not imply that an association learned from missingness survives an arbitrary change in recording conditions.

Missingness-shift risk has already been studied directly. Rockenschaub and colleagues established conditions for invariance of optimal predictors and showed that exploiting outcome-dependent missingness can improve source performance yet harm performance after a shift [4]. Their general probability formulation extends beyond continuous regression, and predictor invariance does not imply unchanged risk. A further complication is that constant imputation can itself preserve information about absence. Thus a model receiving imputed values without explicit flags need not be insensitive to missingness [3, 5]. A comparison with and without flags measures the incremental effect of their explicit representation.

We examine that incremental effect in a controlled seven-class classification problem using real, complete Dry Bean measurements. Artificial omissions change their class association while retaining calibrated expected single-feature omission probabilities. This separates changes in the association from a simple change in the expected amount of missing data, without claiming to hold the joint mask distribution fixed. The principal comparison adds 16 flags to the same imputed values. Outcome-independent training, three omission

probabilities, two fixed learners, a flags-only diagnostic and class-specific recall then test the scope and interpretation of the comparison. The contribution is a transparent finite-sample illustration of a known transfer problem, rather than a new robustness theorem or a model-selection rule for naturally incomplete data.

Materials and methods

Data and analysis population

The public UCI Dry Bean dataset contains 13,611 observations from seven bean classes and 16 numerical measurements derived from bean images [6, 7]. The supplied measurements contain no missing or nonfinite values. We used all 16 features in source order: Area, Perimeter, MajorAxisLength, MinorAxisLength, AspectRation, Eccentricity, ConvexArea, EquivDiameter, Extent, Solidity, roundness, Compactness and ShapeFactor1 through ShapeFactor4. The target classes, in the alphabetical order used by the generator, were BARBUNYA, BOMBAY, CALI, DERMASON, HOROZ, SEKER and SIRA.

Before splitting, we retained the first occurrence of each exact feature vector. This removed 68 duplicate rows, all with the same HOROZ label as their retained counterpart; no duplicate feature group had conflicting labels. The analysis therefore contained 13,543 rows, with class counts of 1,322, 522, 1,630, 3,546, 1,860, 2,027 and 2,636 in the order above. Removing exact duplicates prevented identical feature vectors from entering both training and test sets. No observations were discarded because of subsequently imposed missingness.

Paired split and omission design

We used 12 stratified random 70/30 splits, with seeds 3100 through 3111. Each split contained 9,480 training and 4,063 test observations. All fitted transformations used training data only. Each seed also determined independent training and test arrays of uniform random numbers, which were reused across omission conditions. Within a condition, every representation received exactly the same rows, observed values and masks. A model was fitted once for a given training condition and then evaluated under five test mechanisms. The roles of the experimental factors are summarized in Table 1.

Table 1. Comparison roles and controlled factors. Every representation pair within a condition shares rows, observed values, masks and evaluation labels.

Factor	Levels or definition	Scientific role
Training strength	0 (MCAR), 2 (class coded)	Separate absence of trained mask signal from informative training
Evaluation strength	-2, -1, 0, 1, 2	Reverse, attenuate, erase or retain class association
Omission probability q	0.10, 0.30, 0.50	Test dependence on information loss
Representations	Values; values + flags; flags only	Paired incremental effect and mask diagnostic
Learners	Logistic regression; ExtraTrees	Linear versus nonlinear representation dependence
Complete reference	Unmasked training and test values	Context only; a different information set

Let M = 1 denote omission, c be the class index from 0 to 6, j the feature index from 0 to 15, q the target expected omission probability, and t the strength and direction of the class association. We generated the class-feature probabilities in Equation 1. The intercept a_j was solved separately for every feature, q and t using the training class proportions pi. The expit function maps a real number z to 1/(1 + exp(-z)). Each cell was omitted when its stored uniform draw was below the probability for its class and feature.

$$p_{cj}(q,t) = \operatorname{expit}\left(a_j(q,t) + t\cos\left(\frac{2\pi(c + (j \bmod 7))}{7}\right)\right), \quad \sum_{c=0}^{6} \widehat{\pi}_c p_{cj}(q,t) = q. \tag{1}$$

We considered q = 0.10, 0.30 and 0.50. Training used t = 0 or t = 2, and evaluation used t = -2, -1, 0, 1 or 2. At t = 0, omissions are independent Bernoulli events with probability q, giving the missing-completely-at-random

(MCAR) control. Positive and negative t assign opposing directions to the fixed class code; t = -2 reverses its direction relative to class-coded training, without requiring the probabilities to be exact complements. The arbitrary alphabetical class order and source feature order define this experimental code and carry no biological interpretation. Fig. 1 illustrates the probability construction at q = 0.30.

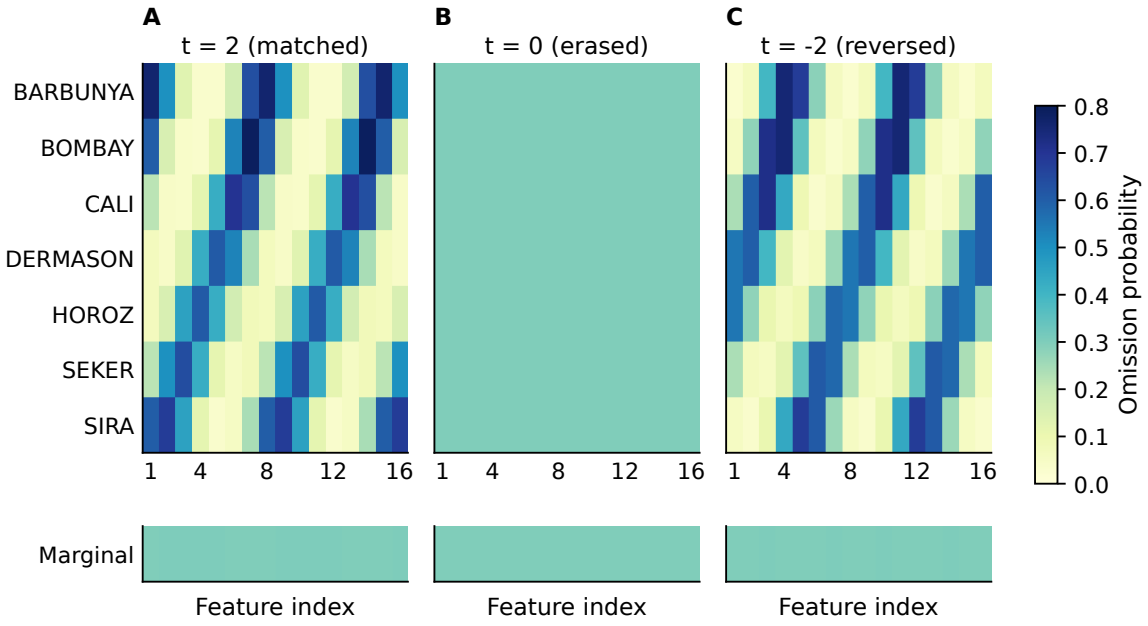


Fig 1. Class-conditional omission can change while expected marginal omission rates remain fixed. A-C: Generator probabilities for q = 0.30 and test strengths 2, 0 and -2, using the training class proportions of repetition 3100. Each cell is P(M = 1 | class, feature), computed from the specified generator, not an empirical estimate of bean measurement failures. Bottom strips show the class-weighted marginal for each feature, equal to 0.30 by construction. Displayed feature indices 1-16 follow the source column order; the formula uses indices 0-15. Test class proportions differ slightly from training proportions, and realized mask rates fluctuate. Labels enter the experimental generator only, not the prediction pipeline.

Calibration makes the expected omission probability of each feature equal to q when averaged over the training class proportions. Realized rates fluctuate, and the test class proportions differ slightly from the training proportions. This constraint does not imply equal joint mask distributions across t. At both training and evaluation, true labels were used only to generate experimental masks. They were never inputs to imputation, scaling or prediction. The study therefore evaluates prediction after a constructed outcome-associated observation process; it does not propose generating deployment masks using unknown labels or establish a naturally occurring bean measurement mechanism.

Representations, learners and complete-data reference

For each corrupted training set, we computed each feature's median from its observed training entries, replaced omitted entries by that median, and fitted a standard scaler to the resulting training matrix. The same medians and scaler were applied to every corresponding test condition. The values representation contained these 16 imputed and standardized features. Values plus flags appended all 16 binary omission indicators, retaining their 0/1 scale. The flags-only representation contained only the indicators and served as a diagnostic of information in the mask. Because constant imputation creates a point mass, the values representation can still convey information about which measurements were absent.

We fitted multinomial logistic regression with L2 regularization, C = 1, the lbfgs solver, a maximum of 1,000 iterations and tolerance 0.000001. The contrasting nonlinear learner was ExtraTreesClassifier with 100 trees, minimum leaf size 3, all features eligible at each split, and one computational worker [8]. Both learners used the split seed as their random state. Settings were fixed throughout; there was no hyperparameter search or

selection based on test performance. The implementation used Python 3.12.4, NumPy 2.2.6, SciPy 1.17.1 and scikit-learn 1.8.0 [9].

For context, we separately fitted each learner on the complete training features and evaluated it on complete test features from the same split, with a train-fitted scaler. This complete-data reference addresses a different information set. It is not an upper bound: class-dependent artificial omissions may provide information about the label in addition to that available in the numerical measurements [5]. No classifier was refitted to exploit knowledge of an evaluation mechanism.

Endpoints and descriptive summaries

The primary endpoint was the paired difference in balanced accuracy between values plus flags and values, at q = 0.30 after training with t = 2, evaluated across the five test strengths. Balanced accuracy was the unweighted mean of recall across the seven classes. We report differences in percentage points, with positive values favouring explicit flags. Secondary endpoints were ordinary accuracy, macro F1, multiclass log loss and multiclass Brier loss. Log loss was the mean negative natural logarithm of the probability assigned to the true class, clipping that probability below at double-precision machine epsilon. Brier loss was the mean sum of squared errors across all seven class probabilities, without division by seven. Smaller probability losses indicate better predictions.

Each reported summary is the arithmetic mean and full observed minimum-maximum range over the 12 split/mask repetitions. Paired-effect ranges summarize within-repetition differences rather than subtracting the extrema of two separate distributions. The test sets overlap across repetitions, and all repetitions reuse one source dataset; these ranges describe computational stability and are not confidence intervals or independent dataset replications. We did not perform significance tests or estimate a practical shift threshold. Class recall was retained for matched and reversed evaluation at q = 0.30 after class-coded training. The complete result tables include all 2,160 model-condition evaluations, including the MCAR-trained and flags-only controls.

Results

A matched gain became a loss when the class association changed

With q = 0.30 and t = 2 in both training and evaluation, explicit indicators improved mean balanced accuracy from 94.63% to 97.77% for logistic regression and from 96.65% to 97.34% for ExtraTrees. The paired gains were 3.14 percentage points [2.59, 3.63] and 0.68 [0.41, 0.98], respectively. Erasing class dependence at evaluation changed the paired effects to -3.94 [-6.76, -2.44] and -7.24 [-8.44, -5.80]. Reversing the code produced larger losses of -20.75 [-24.84, -17.53] and -23.72 [-27.05, -20.36]. Table 2 gives these three directly comparable conditions.

Table 2. Paired effect of adding flags at q = 0.30 after class-coded training. Entries are mean change in balanced accuracy in percentage points [minimum, maximum] across 12 split/mask repetitions; brackets are not confidence intervals.

Learner	Matched (t = 2)	Erased (t = 0)	Reversed (t = -2)
Logistic regression	+3.14 [+2.59, +3.63]	-3.94 [-6.76, -2.44]	-20.75 [-24.84, -17.53]
ExtraTrees	+0.68 [+0.41, +0.98]	-7.24 [-8.44, -5.80]	-23.72 [-27.05, -20.36]

Attenuation did not affect the learners identically. At test t = 1, logistic regression retained a 3.48-point gain [2.67, 4.71], slightly exceeding its mean gain under the exactly matched mechanism. ExtraTrees instead had a small mean loss of -0.33 points [-1.44, 0.24]. At t = -1, the corresponding losses were -16.47 [-19.28, -13.19] and -18.39 [-20.80, -16.46]. The full primary sequence and the other omission probabilities appear in Fig. 2. Thus the sign and size of the incremental indicator effect depended on the test association and learner, rather than solely on the nominal missing rate.

The deterioration was not confined to class-balanced scoring. At full reversal, adding flags changed ordinary accuracy by -30.98 percentage points [-33.97, -27.94] for logistic regression and -27.71 [-29.90, -25.62] for ExtraTrees. Macro F1 differences were -0.200 [-0.238, -0.170] and -0.232 [-0.262, -0.198], respectively. These

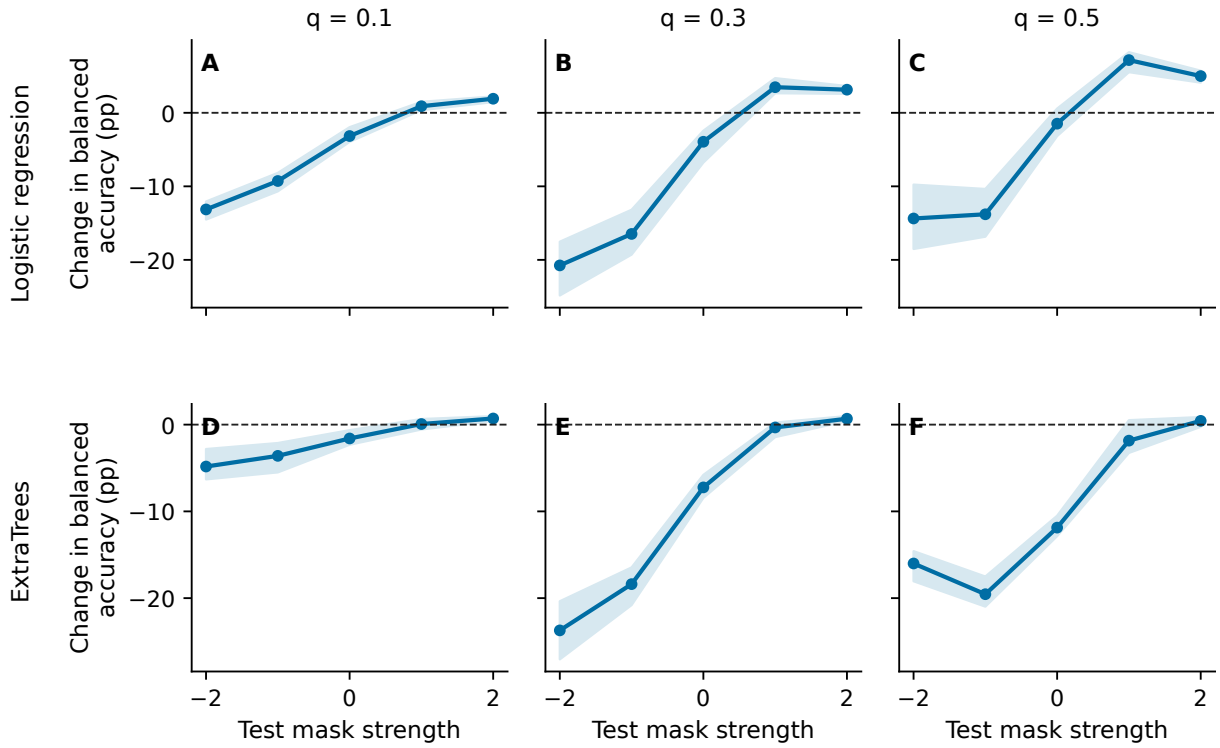


Fig 2. The incremental value of explicit indicators depends on the evaluation mask relation. Values-plus-flags minus values-only balanced accuracy, in percentage points, after training with strength 2. Columns: q = 0.10, 0.30 and 0.50. A-C: Logistic regression. D-F: ExtraTrees. Points and lines are means over 12 paired stratified split/mask repetitions; shading spans their full observed minimum-maximum range. The same observed values and masks are used by both representations within each condition. Test strength 2 matches training, 0 erases class dependence and -2 reverses the code. Vertical scales are shared within each model row. These ranges describe computational stability and are not confidence intervals over independent datasets.

endpoints agree on the direction of the primary loss, while their different magnitudes reflect the unequal class frequencies and the distribution of errors.

Absolute performance and mask-only predictions expose the information change

The values representation also deteriorated as the class code changed. At q = 0.30, its logistic-regression balanced accuracy fell from 94.63% [94.13, 95.17] under matching to 47.68% [44.99, 50.27] under reversal; values plus flags fell from 97.77% [97.44, 98.32] to 26.92% [21.24, 31.54]. ExtraTrees values fell from 96.65% [96.24, 97.51] to 57.48% [54.06, 63.30], and values plus flags from 97.34% [96.87, 97.99] to 33.76% [29.43, 37.31]. The paired loss attributable to adding flags is therefore one part of a broader change in the information available after masking. Fig. 3 places these absolute performances beside the mask diagnostic and probability losses.

Indicators alone predicted the matched artificial class code: balanced accuracy was 65.27% [63.82, 66.52] for logistic regression and 58.06% [57.06, 59.66] for ExtraTrees. After reversal, the same fitted diagnostics achieved only 0.34% [0.23, 0.43] and 0.77% [0.43, 1.16]. Under erased class dependence, their scores were 13.90% [13.27, 14.36] and 14.10% [13.02, 14.88], near the one-seventh balanced-accuracy reference. This establishes that the constructed masks contained learnable class information and that reversing their relation to class made the learned code actively misleading. It does not quantify how much of either combined learner's prediction was caused by any individual flag.

Probability losses showed substantial deterioration beyond the change in predicted classes. Logistic-regression log loss was 0.176 [0.158, 0.188] for values and 0.069 [0.059, 0.079] for values plus flags

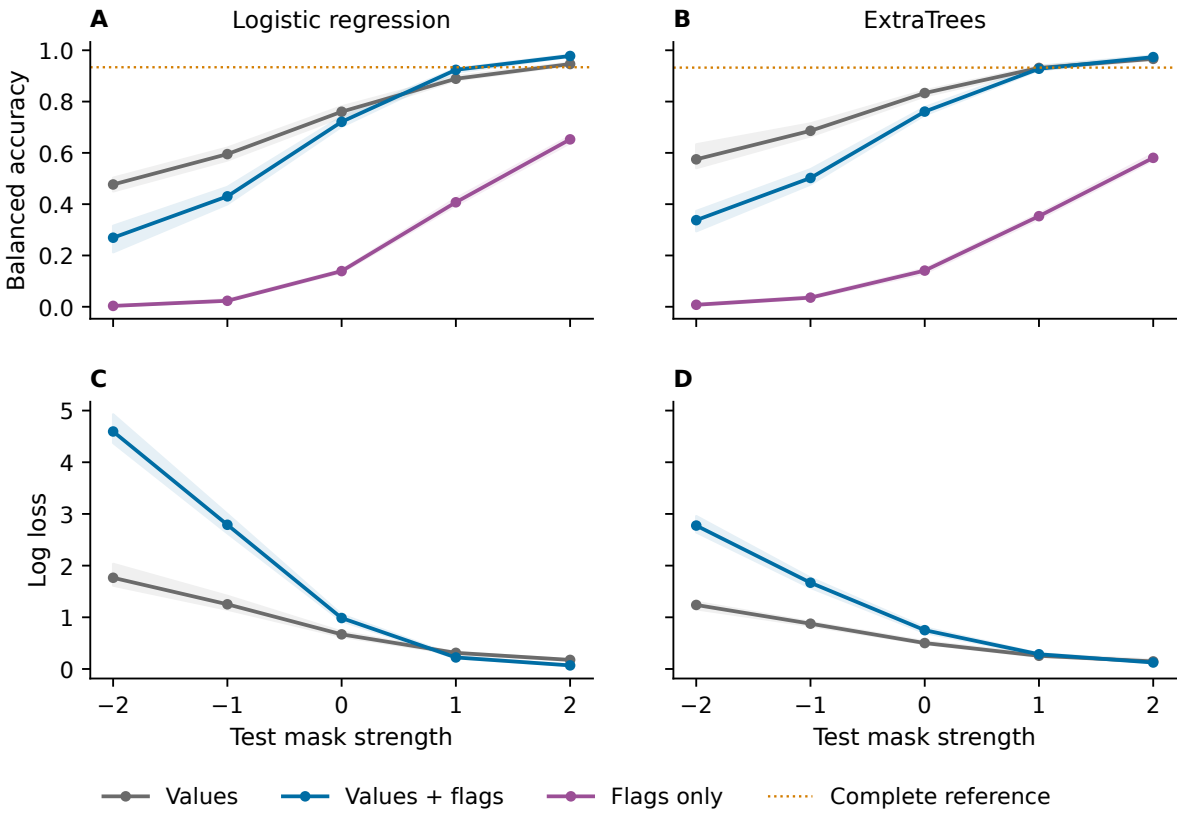


Fig 3. Absolute performance reveals information in the mask and loss of transfer. All models use q = 0.30 and training strength 2. A-B: Balanced accuracy for imputed values, the same values plus flags, and flags alone. C-D: Log loss for values and values-plus-flags, where smaller is better; the flags-only diagnostic is omitted here because near-zero probabilities dominate its loss (retained in the complete results table). Left: Logistic regression. Right: ExtraTrees. Lines are repetition means and shading spans the full range across 12 split/mask repetitions; these are not confidence intervals. Dotted lines in A-B show separately fitted complete-data references on complete test features. This reference is not an upper bound because artificial masks can encode class information absent from the complete-feature task. Values-only models can also recover some mask information from the imputation point mass.

under matching, versus 1.765 [1.614, 2.033] and 4.595 [4.374, 4.927] under reversal. ExtraTrees had matched losses of 0.146 [0.133, 0.181] and 0.125 [0.109, 0.142], rising to 1.239 [1.157, 1.297] and 2.775 [2.645, 2.958]. The reversal increased the paired Brier loss by 0.607 [0.544, 0.678] and 0.324 [0.304, 0.348], respectively. Flags-only tree log losses were much larger and sensitive to probabilities at or near zero; their complete values are retained in the supporting results rather than compressed onto the main loss panels.

Complete-data reference balanced accuracy was 93.38% [92.66, 94.25] for logistic regression and 93.24% [92.84, 93.61] for ExtraTrees. Both values-plus-flags models exceeded these reference means under matched q = 0.30 masks. Because the artificial mask supplied class-related information, that comparison does not demonstrate that removing measurements improves an otherwise unchanged prediction problem. It instead shows why the unmasked reference cannot be treated as a ceiling for this construction.

Omission probability and training mechanism bounded the finding

Changing q preserved the contrast between matched benefit and reversal loss but changed its magnitude. With class-coded training at q = 0.10, matched gains were 1.91 points [1.46, 2.23] for logistic regression and 0.71 [0.42, 0.97] for ExtraTrees; reversal losses were -13.14 [-14.51, -12.07] and -4.83 [-6.31, -2.82]. At q = 0.50, matched effects were 5.00 [4.11, 5.72] and 0.43 [-0.26, 0.87], while reversal effects were -14.37 [-18.52, -9.77] and -16.01 [-18.05, -14.59]. The high-q tree gain crossed zero across repetitions, and the most negative relative

effect did not increase monotonically with q or reversal strength. At q = 0.50, ExtraTrees had a larger relative loss at t = -1 than at t = -2, where both representations already performed poorly.

MCAR training gave a different pattern. At q = 0.30, flags improved logistic-regression balanced accuracy by 4.18 points [2.99, 4.90] under MCAR evaluation and by 3.80 [2.02, 5.10] under t = -2 evaluation. Across its five test mechanisms, the mean gains ranged from 3.59 to 4.18 points. ExtraTrees increments were much smaller: 0.01 [-0.53, 0.39] for MCAR evaluation and 0.38 [-0.09, 1.11] for t = -2. The corresponding logistic-regression gain under MCAR training and evaluation increased from 0.95 points [0.26, 1.42] at q = 0.10 to 8.47 [7.56, 9.59] at q = 0.50. Indicators therefore had predictive value even when training masks were generated independently of class. The severe reversal losses after class-coded training cannot be generalized to every use of indicators.

The expected feature-wise calibration error was below 0.0000000000001. Realized aggregate test omission fractions across conditions and repetitions ranged from 0.0972 to 0.1031 at q = 0.10, from 0.2955 to 0.3031 at q = 0.30, and from 0.4968 to 0.5048 at q = 0.50. The marked changes in performance therefore occurred while the expected feature-wise omission rates were controlled and realized overall rates remained near their targets. These observations do not establish equality of higher-order mask patterns.

Class-specific recall showed unequal consequences

The aggregate loss hid differences between classes and learners. At q = 0.30 under reversal, logistic regression lost 76.52 recall points [-79.14, -74.84] for SIRA after adding flags, but gained 30.40 [17.95, 44.87] for BOMBAY. ExtraTrees lost a mean 53.82 points [-61.44, -39.32] for SIRA and 8.49 [-19.87, 4.49] for BOMBAY. Under matching, logistic regression gained mean recall in every class, whereas the ExtraTrees BOMBAY effect was -0.32 points [-1.92, 0.64] despite its positive aggregate effect. Fig. 4 shows all seven classes for the two endpoint mechanisms.

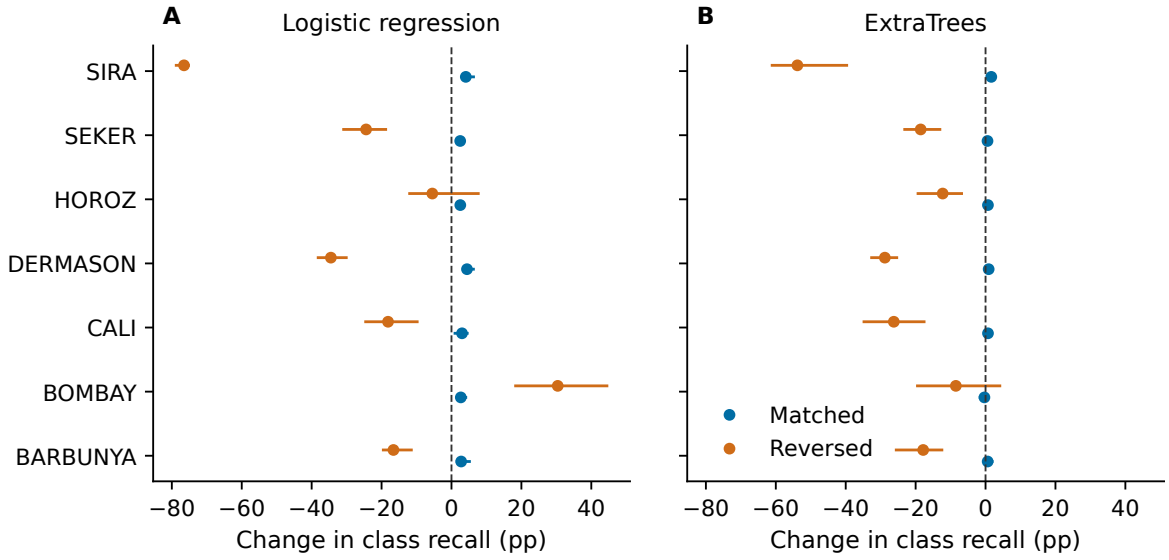


Fig 4. Class-specific effects of explicit flags under matched and reversed masks. Values-plus-flags minus values-only recall, in percentage points, at q = 0.30 after training with strength 2. A: Logistic regression. B: ExtraTrees. Points are means and horizontal segments span the observed range over 12 paired split/mask repetitions. Each class uses its own held-out denominator, retained in the class-recall table. Class names identify source labels and do not imply a naturally observed class-dependent measurement mechanism. Intervals are descriptive repetition ranges.

Class test denominators were 397, 156, 489, 1,064, 558, 608 and 791 in alphabetical order in every repetition. These denominators explain why ordinary accuracy and balanced accuracy weight the consequences differently. The heterogeneous responses also show that reversal is not a uniform subtraction from every class's recall. Since the generator assigned the code using an arbitrary class order, these differences describe this controlled classification construction rather than class-specific propensities for real bean measurements to fail.

Discussion

The experiment separates two questions that a matched train-test comparison alone cannot answer. Explicit indicators improved classification under the matched class-coded mechanism, but the same fitted models lost that incremental advantage when the association was erased or reversed. The mask-only diagnostic confirms that the construction created a strong learnable class signal. Together, these results show why predictive value under one observation process is conditional on how that process transfers. Controlling expected per-feature omission rates leaves substantial freedom for the class information carried by absence to change.

This finding agrees with the nearest missingness-shift work rather than overturning it. Rockenschaub and colleagues already demonstrated the vulnerability of outcome-associated missingness and distinguished assumptions for predictor invariance from equality of risk [4]. Our generator does not impose their ignorability conditions, and finite logistic regression and ExtraTrees are not Bayes-predictor oracles. The present evidence adds a paired multiclass illustration with fixed expected feature-wise margins, contrasting explicit representation against the same imputed numerical values. It cannot establish a general failure of methods that use missingness or invalidate an invariance result whose assumptions it does not test.

The MCAR-trained controls also prevent a simple interpretation that indicators are useful only because they memorize a class-dependent recording policy. Their gains for logistic regression were substantial even when the training mask had no designed class association. Flags can distinguish an imputed value from a measured value and allow the learner to accommodate the representation produced by imputation. This explanation is compatible with the results, but the experiment does not isolate its contribution from other finite-sample representation effects. The small tree increments in the same controls reinforce the need to distinguish learners. The broadly favourable within-dataset findings of Lenz and colleagues and the conditional deployment findings of Sisk and colleagues address related questions under different observation assumptions [2, 3].

Removing explicit flags is consequently an incomplete safeguard. The values models also deteriorated under class-association reversal, and constant imputation can reveal absence through its point mass [3, 5]. In addition, changing which measurements are observed changes the available covariate information even if the expected omission frequency is fixed. Our comparison cannot attribute all values-model deterioration to implicit mask recognition. It establishes the incremental effect of adding flags within a condition, while absolute performance reflects both representation and the observation process. The flags-only results and complete-data references help interpret that distinction without turning either into an optimal comparator.

For applied evaluation, the implication is to examine plausible changes in the relation between missingness and outcomes alongside the usual matched evaluation. A stable overall missing percentage cannot, by itself, validate that relation. Probability scores and class-specific recall can reveal consequences that a single accuracy number obscures. The present design demonstrates the value of these comparisons, but does not identify a practically tolerable value of t, a monitoring rule, or a universally robust training strategy. MCAR training was informative as a control; its performance here does not establish that artificially randomizing omissions is an appropriate remedy in deployment.

Several limits determine the scope of these conclusions. The study uses one public dataset with correlated morphological measurements, two untuned learners and a single median-imputation pipeline. Every missing entry is artificial; no naturally occurring omissions, temporal changes, sites or prospective deployments were observed. The sinusoidal class code and its strength are experimental choices, and alternative codes or dependence between omitted cells could produce different quantitative effects. Calibration fixes expected single-feature margins, not realized margins, the joint mask distribution or the amount of class information. Twelve overlapping split/mask repetitions describe sensitivity within this dataset and do not provide population confidence intervals. Broader datasets and observed recording-process changes are needed to assess how often the illustrated failure matters in practice.

Conclusions

On the evaluated Dry Bean classification task, explicit missingness indicators improved prediction under matched class-dependent omissions but produced large incremental losses when that association reversed. Their usefulness under MCAR training and the deterioration of values-only models show that the relevant issue extends beyond whether flags are present. Predictive information in both the observation process and its representation must transfer. Evaluating that transfer requires contrasts that vary the association, inspect absolute performance and retain class-specific consequences, even when expected marginal missing rates remain stable.

Data and code availability

The original Dry Bean data are available from the UCI Machine Learning Repository under DOI 10.24432/C50S4B [6]. The accompanying local research package contains the source data copy, retained source-row indices, experiment code, split definitions, complete evaluation tables, per-case predictions and losses, and editable figure and manuscript sources. The supplied code records the seeds and settings needed to reproduce the omission construction and model comparisons.

Ethics statement

This study reanalysed public numerical measurements of dry beans. It collected no new human-participant or animal data.

Supporting information

S1 Table. Complete model-condition summaries. File S1_Table.csv gives the mean and full minimum-maximum range of balanced accuracy, accuracy, macro F1, log loss and Brier loss over 12 split/mask repetitions for all 180 experimental combinations. Ranges are descriptive, not confidence intervals.

S2 Table. Paired effects of explicit indicators. File S2_Table.csv gives all 60 condition-specific values-plus-flags minus values contrasts, summarized across the same repetitions. Positive changes favour flags for the classification scores and disfavour flags for the probability losses.

S3 Table. Class-specific recall and denominators. File S3_Table.csv retains recall for each class, learner, representation and repetition under matched and reversed evaluation at omission probability 0.30 after class-coded training, with the number of test observations in each class.

S4 Table. Per-repetition model scores. File S4_Table.csv contains all 2,160 learner-representation-condition evaluations, including split identifiers, training and test sample counts, and all five prediction endpoints.

S1 Archive. Data, code and editable sources. File S1_Archive.zip provides the UCI source data copy and attribution, retained source-row indices, split definitions, per-case predicted labels and losses, complete result files, producing scripts, editable figures, manuscript sources and instructions for recomputing summaries, rerunning the experiment and rebuilding the manuscript.

References

1. Josse J, Chen JM, Prost N, Varoquaux G, Scornet E. On the consistency of supervised learning with missing values. Statistical Papers. 2024;65(9):5447-79. Available from: https://arxiv.org/abs/1902.06931. doi:10.1007/s00362-024-01550-4.
2. Sisk R, Sperrin M, Peek N, van Smeden M, Martin GP. Imputation and missing indicators for handling missing data in the development and deployment of clinical prediction models: A simulation study. Statistical Methods in Medical Research. 2023;32(8):1461-77. doi:10.1177/09622802231165001.
3. Lenz OU, Peralta D, Cornelis C. No Imputation without Representation; 2023. Data-centric Machine Learning Research Workshop at ICML 2023. Available from: https://dmlr.ai/assets/accepted-papers/87/CameraReady/20230701_missing-indicators.pdf.
4. Rockenschaub P, Xian Z, Zamanian A, Piperno M, Ciora OA, Pachl E, et al.. Robust prediction under missingness shifts; 2024. arXiv preprint arXiv:2406.16484v1. Available from: https://arxiv.org/abs/2406.16484. arXiv:2406.16484.
5. Albu E, Gao S, Wynants L, Van Calster B. missForestPredict—Missing data imputation for prediction settings. PLOS ONE. 2025;20(11):e0334125. doi:10.1371/journal.pone.0334125.
6. Dry Bean [Dataset]; 2020. UCI Machine Learning Repository. Available from: https://archive.ics.uci.edu/dataset/602/dry+bean+dataset. doi:10.24432/C50S4B.

7. Koklu M, Ozkan IA. Multiclass classification of dry beans using computer vision and machine learning techniques. Computers and Electronics in Agriculture. 2020;174:105507. doi:10.1016/j.compag.2020.105507.

8. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Machine Learning. 2006;63:3-42. doi:10.1007/s10994-006-6226-1.

9. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12:2825-30. Available from: https://www.jmlr.org/papers/v12/pedregosa11a.html.