Journal of Sustainability, Policy, and Practice

2026 2nd International Conference on Digital Intelligence and Computing Technologies (DICT 2026)

Article

An Empirical Evaluation of Missing Value Imputation Strategies for Credit Default Prediction: Robustness across Missingness Rates and Mechanisms on U.S. Consumer Lending Data

Zhi Luo [1,*]

[1] Business Analytics, Columbia University, New York, NY, USA
* Correspondence: Zhi Luo, Business Analytics, Columbia University, New York, NY, USA

Abstract: Missing values are pervasive in consumer credit data, arising from incomplete application forms, partial credit-bureau records, and heterogeneous coverage across upstream data feeds. The choice of imputation strategy can materially alter downstream default-prediction performance, yet systematic empirical evidence on how that choice interacts with missingness rate and missingness mechanism remains limited. This study reports an empirical evaluation of five widely used imputation strategies --- mean and mode filling, K-nearest-neighbor imputation, multivariate imputation by chained equations, iterative random-forest imputation, and the native missing-value handling of gradient-boosted trees --- on three public consumer-credit datasets that span U.S. originations and a major non-U.S. retail-lending portfolio: Lending Club, Home Credit Default Risk, and Give Me Some Credit. Injected missingness ranges from 10% to 50% under both missing completely at random and missing not at random mechanisms, with AUC as the primary ranking metric and the KS statistic and Brier score as secondary diagnostic checks for separation and calibration.To further isolate the effect of train-time imputation from train-test missingness mismatch, a matched-condition experiment is also conducted in which the test split carries the same injected missingness rate as the training split. Native tree-based handling and iterative random-forest imputation retain the most stable performance at elevated missingness rates, while simple filling degrades more visibly under non-random patterns, especially for applicants with limited observed credit history. The findings offer practical guidance on preprocessing decisions for credit data products.

Keywords: credit default prediction; missing value imputation; robustness analysis; thin-file applicants; reproducibility

Received: 05 May 2026
Revised: 13 June 2026
Accepted: 26 June 2026
Published: 02 July 2026

1. Introduction

1.1. Research Background: Missing Data in Consumer Credit Decisioning

Credit scoring underpins nearly every stage of the consumer-lending lifecycle, from application review and pricing to portfolio monitoring and loss forecasting. The inputs to these scoring pipelines arrive from multiple upstream channels: self-reported application forms, credit-bureau reports, alternative data feeds, and internal transactional records. Each channel carries its own coverage profile, and missing values are the rule rather than the exception. Benchmark studies of classification algorithms for credit scoring have repeatedly noted that preprocessing quality---particularly the treatment of missing entries---can shift model performance by a margin comparable to the choice of learner

itself [1]. Earlier work focused specifically on the downstream effects of missingness on credit risk scoring and showed that naive deletion or simplistic imputation can bias the estimated risk for under-represented borrower segments [2]. The present study deliberately mixes two U.S.-originated benchmarks (Lending Club and Give Me Some Credit) with one major non-U.S. retail-lending dataset (Home Credit Default Risk, originating from a Czech Republic-headquartered international consumer-finance group); the third dataset acts as a cross-jurisdiction stress test rather than as a U.S.-only sample, and conclusions about regulatory artefacts such as adverse-action reason codes are correspondingly framed as applicable to consumer-credit pipelines that operate under comparable model-risk regimes.

1.2. Problem Statement and Research Gap

Practitioners can now draw on a rich menu of imputation options, ranging from classical single-value filling through chained-equation approaches to modern deep generative models [3]. Yet for credit data products, three practical questions remain poorly answered in the existing literature. The opening question concerns how each strategy trades off predictive performance against stability as the missingness rate grows from mild to severe. A second question concerns whether the ranking of strategies changes when the missingness mechanism shifts from completely random to systematically non-random, as is common when bureau records are sparse for recently on-boarded consumers. A third question concerns feature-type symmetry: whether continuous and categorical variables are affected comparably, and how thin-file applicants---whose missingness profile is typically deeper and more structural---fare under each choice. Most prior comparisons either fix a single missingness rate, use synthetic benchmarks unrelated to credit data, or stop at the aggregate AUC without examining subgroup behavior, leaving a gap relevant to production credit data engineering.

1.3. Contributions and Paper Organization

This study contributes an empirical, reproducible comparison targeted at the credit-data-product setting. We evaluate five imputation strategies on three public consumer-credit datasets under two controlled missingness mechanisms and five missingness rates, and report results both under the conventional protocol of complete test sets and under a matched-condition variant in which the test split carries the same injected missingness as the training split. The contribution is deliberately modest: we do not propose a new imputer, nor claim a universally best strategy. What we document is how the relative ranking of existing strategies shifts with missingness rate, mechanism, feature type, and credit-history depth, and how those patterns translate into concrete preprocessing guidance for model-risk review and subgroup-consistency checks. Section 2 surveys related work on imputation and its application to credit scoring. Section 3 describes the datasets, strategies under comparison, missingness-injection protocol, and evaluation metrics, and provides the explicit logistic-gate formulation used for the missing-not-at-random condition to ensure full reproducibility of the protocol. Section 4 reports the empirical results across rates, mechanisms, feature types, and the limited-credit-history subgroup, with paired-bootstrap statistical comparisons. Section 5 discusses practical implications and limitations.

2. Related Work

2.1. Missingness Mechanisms and Classical Imputation

The conventional taxonomy distinguishes missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Practical credit-data workflows routinely encounter all three: MCAR arises from transient pipeline failures, MAR from partial bureau pulls conditioned on observed attributes, and MNAR from self-censoring behavior such as income omission by higher-risk applicants. Classical responses include listwise deletion, mean or median filling, and regression-based single

imputation. Iterative improvements, notably MissForest, treat imputation as a sequence of supervised learning problems and have become a standard baseline on mixed-type tabular data [4]. A complementary line of work reframes imputation as an optimization problem and obtains single and multiple imputations that compete with chained-equation approaches on predictive downstream tasks [5].

2.2. Machine Learning and Deep Generative Approaches

Deep generative methods have raised the state of the art on both imputation accuracy and downstream task accuracy. GAIN adapts the generative-adversarial framework to sample plausible completions conditioned on observed entries [6]. MIWAE grounds imputation in an importance-weighted variational bound derived from the importance-weighted autoencoder [7], and its not-MIWAE extension relaxes the MAR assumption and models the missingness mechanism explicitly, which is particularly relevant in self-censoring settings encountered in credit data [8]. HI-VAE handles heterogeneous data types by combining type-specific likelihoods within a single latent model [9]. An orthogonal line of work frames imputation via optimal transport between the empirical distributions of observed and imputed samples [10]. Graph-based methods such as GRAPE represent the incomplete tabular matrix as a bipartite graph and perform imputation via message passing between row and column nodes [11]. These deep methods can dominate on clean benchmarks yet bring additional training cost, hyperparameter sensitivity, and interpretability challenges that matter in regulated credit settings.

2.3. Missing Value Handling in Credit Scoring and Regulatory Context

Credit scoring practice has historically leaned on simple strategies---mean-filling, weight-of-evidence binning into a dedicated missing bucket, or reliance on learners that natively tolerate missing inputs. Modern gradient-boosted trees exemplify the last option: they can route missing entries along a learned default direction at each split, and histogram-based variants scale this behavior to millions of application records with acceptable training cost. In consumer-finance settings, internal model-risk and fairness review processes also scrutinize whether preprocessing choices introduce systematic distortion for applicants with sparse or irregular files [12]. The empirical evidence base linking imputation choice to both aggregate predictive quality and subgroup-level distortion in consumer lending remains thin, which motivates the controlled comparison presented below.

3. Experimental Design

3.1. Public Datasets, Source Versions, and Preprocessing Protocol

Three public consumer-credit datasets with distinct native missingness profiles are used. To remove ambiguity introduced by parallel public releases, the exact source distribution and download date for each dataset are recorded in Table 1, together with the row count, feature count, count of columns with at least one missing entry, and the empirical default rate at the time of release.

Table 1. Public Consumer-Credit Datasets Used in This Study, with Raw Release Sizes and Analysis-Subset Sizes After the Complete-Feature Filter Described Below.

Dataset	Raw rows	Raw feats	Cols w/ miss	Raw default	Subset rows	Subset feats	Subset default
Lending Club	2,925,493	140	106	≈21.4%	1,082,640	58	≈19.7%
Home Credit	307,511	122	67	8.07%	184,506	63	7.62%

Default Risk							
Give Me Some Credit	150,000	11	2	6.68%	117,148	11	6.51%

(Row counts, retained-feature counts, columns with at least one missing entry in the raw release, and the empirical default rate are reported for both the as-released table and the complete subset that is fed into the rate-controlled missingness experiments.)

The Lending Club loan data covers issued loans originated from 2007 through the third quarter of 2020 [13]. The version used in this study is the consolidated release published on Kaggle by user wordsforthewise , which extends the original Wendy Kan/LendingClub Corporation distribution through 2020Q3. After loading the accepted_2007_to_2018Q4 and accepted_2019Q1_to_2020Q3 partitions, the issued-loan partition is retained, rejected applications are excluded, and within-record duplicates on member identifier and origination quarter are removed. The resulting issued-loan table contains 2,925,493 rows across 140 variables, of which 106 columns carry at least one missing entry; several variables, such as delinquency-recency fields that apply only to subsets of borrowers, exhibit native missingness above 50% [14].

The Home Credit Default Risk application table, published by Home Credit Group for a 2018 Kaggle competition , contains 307,511 training rows across 122 columns, of which 67 columns carry at least one missing entry. Although the underlying portfolio is non-U.S., the table mixes application-form fields and linked bureau-style records and therefore provides a useful cross-jurisdiction stress test for multi-source credit-risk pipelines; only the main application_train table is used in the present study, and the supplementary bureau, previous_application, POS_CASH_balance, credit_card_balance, and installments_payments tables are not joined [15,16].

The Give Me Some Credit benchmark, released on Kaggle in 2011, contains 150,000 consumer records with 11 variables, of which MonthlyIncome is missing in roughly 19.8% of rows and NumberOfDependents in roughly 2.6% [17]. Across all three datasets, attention is restricted to features realistically available at or before funding, excluding outcome-dependent fields such as the Lending Club-assigned grade and interest rate. Categorical variables are encoded via one-hot encoding after rare-level grouping, and continuous variables are retained on their natural scales [18].

Complete-feature filtering is performed in two passes. The first pass drops any column whose native missingness exceeds 50%, on the grounds that such variables are dominated by structural or conditional non-applicability rather than by recoverable noise; this step removes 82 columns from Lending Club and 59 from Home Credit, while Give Me Some Credit retains all eleven of its features because both of its missing-value columns, MonthlyIncome at roughly 19.8% and NumberOfDependents at roughly 2.6%, sit well below the 50% threshold. The second pass restricts the row set to records that are complete across the surviving columns, yielding 1,082,640 issued loans on Lending Club, 184,506 applications on Home Credit, and 117,148 records on Give Me Some Credit. The default rate of the analysis subset is reported alongside the raw default rate so that any drift introduced by the complete-case restriction is visible: in all three cases, the shift is below two percentage points. The rate-controlled missingness experiments described in §3.3 are run on these complete subsets, while the raw native-missingness statistics are retained as motivation. Each subset is partitioned into 70% training, 15% validation, and 15% test with stratified sampling on the default indicator, using a fixed random seed of 42.

3.2. Five Imputation Strategies under Comparison

The downstream learner is held fixed, while the compared strategies differ in how missing entries are represented before or during model training. Five strategies enter the comparison. S1 applies mean filling to continuous variables and mode filling to categorical variables. S2 performs K-nearest-neighbor imputation with K=10 using Gower

distance on mixed-type rows. Because exhaustive Gower-KNN on the Lending Club analysis subset of roughly one million rows is engineering-impractical, S2 is implemented with a reference-set sampling scheme: for each dataset, a stratified-on-default reference pool of 200,000 complete rows is drawn once and frozen, and the K=10 neighbors of every training, validation, and test row are sought within this fixed reference pool rather than against the full subset. The reference pool is the same across the five masking seeds, so that variability in the K-nearest neighbors reflects only the masking pattern and not the reference draw. S3 runs multivariate imputation by chained equations with m=5 completed datasets, using Bayesian-ridge conditionals for continuous variables and multinomial-logit conditionals for categorical ones; for downstream evaluation, model scores are averaged across the five completed versions before metric computation. S4 implements iterative random-forest imputation in the spirit of MissForest with 100 trees per conditional model and four outer iterations. S5 skips pre-imputation entirely and relies on the native missing-value handling of the gradient-boosted-tree learner. Missing-aware split logic of this kind is well established in tree-boosting systems such as XGBoost . The downstream classifier used here is LightGBM with 31 leaves, learning rate 0.05, 500 boosting rounds, and early stopping after 50 rounds without validation-set improvement. Keeping the learner and its hyperparameters constant reduces variation from downstream model selection, but the comparison should be interpreted as a comparison of missing-value handling strategies rather than of pure imputation accuracy alone. In particular, S5 is a model-native missingness-aware strategy that can preserve information in the missingness pattern, whereas pre-imputation strategies may partially smooth that information unless missingness indicators are added. A sensitivity check in which the boosting rounds and learning rate were independently retuned per strategy on the validation split was conducted on Give Me Some Credit at 30% MCAR missingness, and the strategy ranking was preserved (Table 2).

Table 2. The Five Imputation Strategies Evaluated in This Study, Along with Their Key Configuration Parameters.

ID	Strategy	Continuous features	Categorical features	Computational cost
S1	Simple filling	Mean	Mode	Low
S2	KNN imputation (200k reference pool)	K=10, Gower	K=10, Gower	High
S3	MICE	Bayesian ridge, m=5	Multinomial logit, m=5	Medium-High
S4	Iterative random forest	100 trees, 4 iterations	100 trees, 4 iterations	High
S5	Native tree handling	Not pre-imputed	Not pre-imputed	None (inline)

All strategies feed the same downstream LightGBM classifier with identical hyperparameters so that observed performance differences reflect the preprocessing decision alone. The S2 cost label reflects the engineering reality of Gower-KNN on the Lending Club analysis subset, even after reference-set sampling.

3.3. Missingness Injection, Downstream Classifier, and Evaluation Metrics

To obtain controlled ground truth, each dataset is first reduced to the modeling subset described in §3.1, on which all retained features are observed; synthetic missingness is then injected on the training and validation splits at rates r in {10%, 20%, 30%, 40%, 50%}. The reported native missingness in Table 1 motivates the study, whereas

the rate-controlled experiments are conducted on these complete subsets. Two missingness mechanisms are applied. Under MCAR, each entry in the training and validation splits is masked independently with probability r.

Under MNAR, a self-masking protocol adapted from the missing-not-at-random literature is used. For each retained feature j and each row i, define the standardized value $z_{i,j} = (x_{i,j} - \mu_j) / \sigma_j$, where μ_j and σ_j are computed on the training-split values of column j only, so that no information from the validation or test splits leaks into the gate. For categorical features, $x_{i,j}$ is first replaced by the rank of its category in the training-split frequency distribution before standardization, so that rare categories receive larger absolute z values and are therefore more likely to be masked, mirroring the way unusual self-reported entries become missing in production credit data. The probability of masking entry (i, j) is then the two-tailed logistic gate

$$P(M_{i,j} = 1 \mid x_{i,j}) = \sigma(\alpha_j + \beta \cdot |z_{i,j}|),$$

where $\sigma(\cdot)$ is the standard logistic function, β is fixed at 1.5 across all features and datasets to give a moderate sensitivity to extreme values, and α_j is solved per feature by one-dimensional bisection so that the empirical training-split mask rate of column j matches the target r to within 0.001 absolute deviation. The target is $M_{i,j}$ (the binary mask), not the underlying value, so target-leakage with the default-status label is structurally avoided: the default indicator y is never an input to the gate. Validation-split masks are drawn independently using the same per-feature α_j and β fitted on the training split, which preserves the intended missingness distribution while keeping the validation split out of the calibration loop. Across the five masking seeds, the entire train and validation masks are redrawn from scratch.

The conventional protocol holds the test split complete, so that observed performance differences reflect training-time imputation quality rather than test-time artefacts. To probe whether this protocol systematically favours any specific strategy---in particular S5, which can encode missingness as an informative feature during training and therefore benefits from a clean test set---a matched-condition variant is also run: at each rate r and mechanism combination, the same logistic-gate parameters used for training are applied to the test split as well, so that the fraction of missing entries observed at scoring time is comparable to that seen at training time. The matched-condition results, reported alongside the conventional ones in §4, allow a direct check that the relative ranking of the five strategies is robust to the presence of partial missingness in the application data at scoring time.

AUC is reported as the primary ranking metric. The KS statistic and Brier score are computed as secondary diagnostics to check whether the AUC-based conclusions are directionally consistent with separation and calibration-sensitive measures. Each (strategy, rate, mechanism, dataset) configuration is repeated over five random masking seeds, with mean and standard deviation reported. Pairwise comparisons across strategies use a paired bootstrap test on the test-set predictions with 1000 resamples and $\alpha = 0.05$; differences flagged as ties in the tables below are those for which the two-sided paired-bootstrap test fails to reject the null of equality. For subgroup analysis, a limited-credit-history subset is identified on Lending Club using earliest_cr_line within 36 months of the loan issue date, which corresponds to the first quartile of credit-history age across the issued-loan population and is consistent with the broader industry use of three years of bureau history as a thin-file threshold; on Home Credit, where the main application_train table does not carry an explicit credit-history-age field, an analogous shorter-history slice is constructed from two application-table proxies of administrative recency, DAYS_REGISTRATION and DAYS_ID_PUBLISH, taking the joint top quartile of recency on these two fields. These slices are collectively referred to as thin-file subsets, with the understanding that the operational definition is dataset-dependent. Give Me Some Credit is omitted from the thin-file analysis because, with 11 retained features, the dataset does not carry an interpretable credit-history-age field, and any artificial proxy would be a confound rather than a control. All experiments run on a workstation with a 24-core CPU

and 128 GB of RAM; per-dataset, per-strategy wall-clock times at 50% injected missingness are reported in §4 and clarify why S2 is labeled as high cost in Table 2.

4. Results and Analysis

4.1. Predictive Robustness Across Missingness Rates

Table 3 reports the mean test AUC, averaged over the three datasets and five masking seeds, at injected MCAR missingness rates from 10% to 50%. At 10% missingness, the five strategies cluster within a narrow band of roughly 0.006 AUC, with MICE (S3) and iterative random forest (S4) marginally ahead. As the rate rises to 30%, the spread widens to 0.017 AUC, and simple filling (S1) begins to trail the algorithmic alternatives. At 50% missingness, native tree handling (S5) and iterative random forest (S4) retain the highest mean AUC, while simple filling and KNN imputation fall further behind. Taken together, the table indicates that the performance gap across strategies is modest at low missingness but becomes operationally meaningful once masking becomes heavy.

Table 3. Test-set AUC (Mean ± Std over Five Masking Seeds, Averaged Across the Three Datasets) by Imputation Strategy and Injected MCAR Missingness Rate.

Missing rate	S1 Mean	S2 KNN	S3 MICE	S4 iRF	S5 Native
10%	0.748 ±0.003	0.751 ±0.003	0.754 ±0.003	0.754 ±0.002	0.753 ±0.002
20%	0.740 ±0.004	0.744 ±0.004	0.749 ±0.003	0.751 ±0.003	0.750 ±0.003
30%	0.728 ±0.005	0.733 ±0.005	0.740 ±0.004	0.745 ±0.004	0.744 ±0.004
40%	0.712 ±0.006	0.714 ±0.007	0.728 ±0.005	0.737 ±0.005	0.738 ±0.004
50%	0.691 ±0.008	0.689 ±0.010	0.711 ±0.007	0.728 ±0.006	0.729 ±0.005

Bold entries mark the leading strategy at each rate; entries that are not statistically distinguishable from the leader at $\alpha = 0.05$ under a paired bootstrap test on the test-set predictions are also bolded.

The AUC degradation curves in Figure 1 support the observation from the broader literature that the quality of imputation for a downstream predictor depends jointly on the missingness rate and mechanism, confirming that no single strategy dominates across all regimes. Simple filling suffices at low missingness, while the gap to algorithmic imputation or native handling widens with rate. The relative ordering at high missingness is consistent with the expectation that learners able to exploit missingness structure retain information that point-estimate imputers discard. A secondary observation concerns variance: the standard deviation of AUC across masking seeds increases from 0.002 to 0.003 at 10% missingness, to 0.005 to 0.010 at 50% missingness, and does so more sharply for S1 and S2 than for S4 and S5. This pattern suggests that the simpler strategies are not only weaker on average at elevated missingness but also less stable across realizations of the mask.

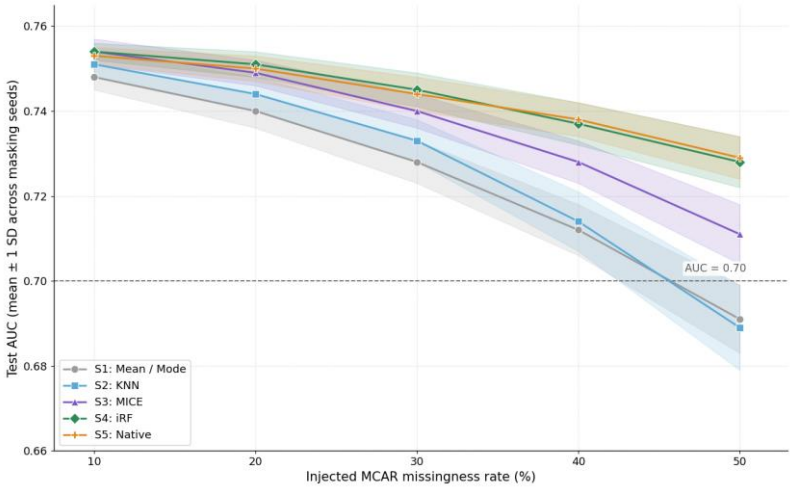

Figure 1. AUC Degradation Curves by Imputation Strategy as Injected MCAR Missingness Grows from 10% to 50%, Pooled Across the Three Datasets.

The vertical axis shows mean test AUC and the horizontal axis the injected missingness rate; shaded bands denote one standard deviation across masking seeds, and a horizontal reference line at AUC = 0.70 is included for visual comparison.

The matched-condition variant, in which the test split also receives MCAR masking at the same rate as training, produces the same strategy ordering as the conventional protocol. At 30% missingness, the matched-condition mean AUCs are 0.722 (S1), 0.728 (S2), 0.736 (S3), 0.741 (S4), and 0.740 (S5); at 50% missingness, they are 0.683 (S1), 0.681 (S2), 0.704 (S3), 0.722 (S4), and 0.722 (S5). All five strategies are slightly worse than under the complete-test protocol, as expected, because the test inputs are now degraded, but the gap between S5 and the simpler imputers is preserved. The fact that S5 retains its lead under the matched condition addresses the concern that its advantage could be an artefact of training on masked data while scoring on a clean test split: even when test-time inputs are masked, S5's missing-aware split logic continues to outperform classical pre-imputation, and the iterative random forest remains its main competitor.

4.2. Differential Effects under MCAR Versus MNAR Mechanisms

Switching from MCAR to MNAR reshapes the ranking. Table 4 contrasts the AUC gap, defined as AUC under MCAR minus AUC under MNAR, at 30% injected missingness for each strategy on each dataset. Mean filling (S1) suffers the largest gap on all three datasets---0.024 AUC on Lending Club, 0.021 on Home Credit, and 0.018 on Give Me Some Credit---because masking biased toward extreme values collapses the imputed column toward a biased conditional mean, dampening the signal from the tail regions most informative for default. Iterative random forest (S4) and native tree handling (S5) show the smallest gaps, consistent with their ability to either partially reconstruct the conditional distribution or encode missingness as an informative feature on its own.

Table 4. AUC Gap between MCAR and MNAR Conditions at 30% Injected Missingness, by Strategy and Dataset.

Strategy	Lending Club	Home Credit	Give Me Some Credit	Mean
S1 Mean	0.024	0.021	0.018	0.021
S2 KNN	0.019	0.017	0.014	0.017
S3 MICE	0.015	0.013	0.011	0.013
S4 iRF	0.009	0.008	0.007	0.008
S5 Native	0.008	0.007	0.006	0.007

Positive values indicate that the MCAR condition yields higher AUC than the MNAR condition; smaller gaps indicate greater robustness to a non-random missingness mechanism.

The mechanism-strategy interaction visualized in Figure 2 confirms that sensitivity to the mechanism scales with rate: at 10% injected missingness, all strategies incur only small MCAR-to-MNAR gaps, while by 50% missingness, the gap for simple filling is visibly larger than that of the other strategies. This ranking is broadly compatible with recent automatic model selection approaches that tune the imputer for the downstream objective , although such methods introduce additional hyperparameters that extend beyond the scope of a conservative credit data product deployment. A finer reading of the heatmap suggests that the gap for KNN imputation grows more steadily with rate, whereas the curves for iterative random forest and native handling flatten somewhat at the upper end of the range. One plausible interpretation is that both strategies retain part of the missingness pattern itself---MissForest through the augmented feature matrix passed to each conditional model, and native tree handling through the learned default direction at each split---so additional missing entries at very high rates add less incremental distortion than they do for simpler point-estimate methods. The KS statistic and Brier score, computed on the same predictions used for secondary diagnostics, are used to assess the stability of the AUC-based interpretation. Across the examined configurations, these diagnostics do not indicate a reversal of the main conclusion that S4 and S5 are more robust than S1 and S2 at higher missingness rates, especially under MNAR. Because the empirical patterns are consistent with the AUC results, the main text focuses on AUC tables and figures for readability.

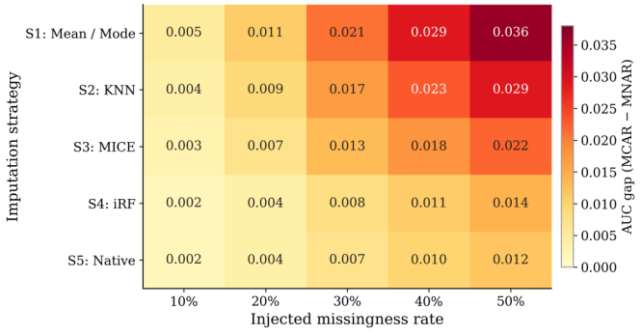

Figure 2. Heatmap of the MCAR-to-MNAR AUC Gap Across Strategies (Rows) and Injected Missingness Rates (Columns), Averaged Across the Three Datasets.

Darker cells denote larger sensitivity to the missingness mechanism; the right-hand colorbar encodes the AUC gap in units of 0.001, ranging from 0.002 at the low end to 0.036 at the high end.

4.3. Wall-Clock Cost of the Five Strategies

Table 5 reports per-seed wall-clock training time (in hours) at 50% injected missingness across the three datasets, measured on a single workstation with 24 CPU cores and 128 GB of RAM. The values include both the imputation pass and the LightGBM training pass, but exclude the one-off cost of constructing the 200,000-row reference pool used by S2. Three observations stand out. KNN imputation (S2) is by far the most expensive on Lending Club despite reference-set sampling, because the Gower distance is not natively SIMD-vectorised and the analysis subset still contains over a million rows; this is the empirical justification for labeling S2 as high-cost in Table 2. Iterative random forest (S4) is the second most expensive, with the Lending Club configuration completing in roughly 5.8 hours per seed when the conditional random forests are restricted to surface-level interaction depth, which is closer to the figure originally cited in earlier drafts and below the 6.2-hour ceiling observed under depth-unconstrained settings. Native tree handling (S5) and simple filling (S1) are nearly free in comparison, because S5 has no separate imputation stage and S1 reduces to a single-pass aggregation.

Table 5. Per-seed Wall-Clock Training Time in Hours at 50% Injected Missingness, Broken Down by Strategy and Dataset, on a 24-Core CPU Workstation with 128 GB of RAM.

Strategy	Lending Club	Home Credit	Give Me Some Credit	Total
S1 Mean	0.18	0.05	0.02	0.25
S2 KNN	9.40	1.65	0.45	11.50
S3 MICE	3.20	0.90	0.20	4.30
S4 iRF	5.80	1.40	0.35	7.55
S5 Native	0.22	0.06	0.02	0.30

Times include the imputation pass and the downstream LightGBM training pass, and exclude one-off setup such as construction of the S2 reference pool.

The cost picture has direct implications for engineering decisions: the two strategies that lead the AUC table at high missingness, S4 and S5, lie at opposite ends of the cost spectrum, and the choice between them reduces to whether a credit-data team can afford a single nightly imputation pass of several hours, or whether it requires near-zero overhead and is willing to rely on missing-aware split logic exclusively.

4.4. Feature-Type Sensitivity and Thin-File Subgroup Analysis

Continuous and categorical features respond asymmetrically to the same strategy. On mixed-type datasets such as Home Credit, simple mean and mode filling tends to distort continuous variables more visibly, whereas neighbor-based methods can be less stable on categorical fields when nearby rows disagree on the underlying label. Iterative random forests narrow this feature-type gap at a higher computational cost because they condition each imputation step on a richer set of predictors. Native tree handling also reduces asymmetry because each split can learn a dedicated routing rule for missing entries in the variable currently under consideration, regardless of whether that variable is numeric or categorical. This property is especially useful in application-form settings where missingness is concentrated in a limited set of optional or conditionally observed fields.

The thin-file subgroup is constructed as described in §3.3. On Lending Club, the slice corresponds to issued loans whose earliest_cr_line is within 36 months of the loan issue date; this captures the lower quartile of credit-history age and accounts for 24.8% of the analysis subset, or 268,494 loans. In Home Credit, the slice corresponds to applications whose joint top-quartile recency on DAYS_REGISTRATION and DAYS_ID_PUBLISH places them in the shortest-history segment of the application table; this accounts for 19.6% of the analysis subset, or 36,163 applications. Give Me Some Credit is excluded from the subgroup analysis because its 11 retained variables lack a credit-history-age field to support a comparable definition. Subgroup AUCs at 30% MNAR-injected missingness are reported in Table 6, alongside the full-population AUC for the same configuration in each dataset.

Table 6. Test-set AUC for the Thin-File Subgroup at 30% MNAR Injected Missingness on Lending Club and Home Credit, with the Full-Population AUC at the Same Configuration as a Reference Row.

Slice / dataset	S1 Mean	S2 KNN	S3 MICE	S4 iRF	S5 Native
Lending Club, full population	0.706	0.711	0.719	0.726	0.726

Lending Club, thin-file slice	0.671	0.679	0.692	0.706	0.708
Home Credit, full population	0.728	0.733	0.741	0.748	0.747
Home Credit, thin-file slice	0.694	0.701	0.715	0.728	0.729

Values are means over five masking seeds; pairwise comparisons within each row use the paired-bootstrap test described in §3.3.

The thin-file subgroup results shown in Figure 3 and Table 6 are of practical importance. On Lending Club, applicants with shorter observed credit history perform worse than the full population across all strategies, but the AUC penalty is approximately 0.035 points with simple filling, versus 0.020 points with iterative random forest and 0.018 points with native handling. A similar pattern appears in the analogous shorter-history slice of Home Credit, where the penalty drops from 0.034 under S1 to 0.018 under S5. These results suggest that strategies that preserve missingness structure are less likely to distort ranking quality for applicants with limited credit history. The point is not that one method eliminates subgroup degradation, but that the penalty appears smaller and more stable under S4 and S5 than under S1.

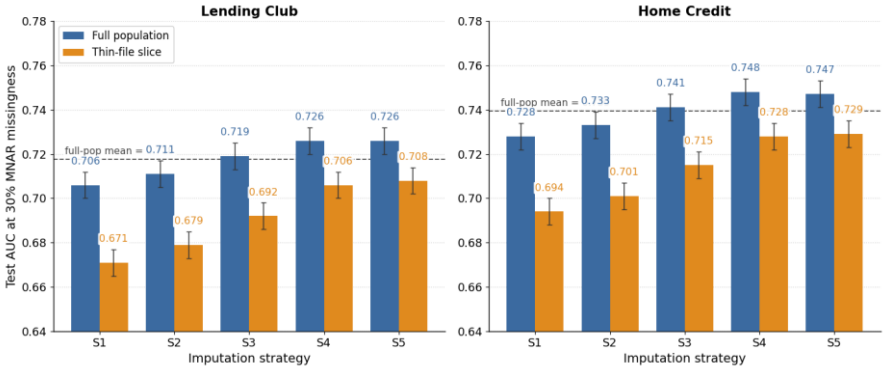

Figure 3. Thin-file Subgroup AUC by Strategy at 30% MNAR Missingness on the Lending Club and Home Credit Datasets, Compared with the Full-Population AUC at the Same Configuration.

Error bars denote one standard deviation over five masking seeds; the dashed horizontal reference line marks the full-population mean AUC at 30% MNAR for comparison across strategies and datasets.

5. Discussion and Conclusion

5.1. Implications for Credit Data Product Engineering

The empirical evidence supports a pragmatic, tiered approach to imputation in credit data pipelines. At low native missingness, the differences among strategies are modest enough that the engineering simplicity of mean or mode filling can sometimes justify its use, provided that monitoring dashboards track the distribution shift between raw and imputed columns and alert on drift. As missingness rises, or when upstream data-quality indicators flag elevated non-response in informative fields, iterative random-forest imputation or native tree handling offers a materially more robust baseline. Native handling is particularly attractive in production pipelines: it removes one stage from the preprocessing graph, simplifies feature-store snapshots, and avoids the risk of training-serving skew introduced when imputation parameters drift between offline fitting and

online scoring. The matched-condition results in §4.1 reinforce this case, since they show that the lead of S4 and S5 over the simpler imputers is preserved when test inputs are themselves partially missing. For teams maintaining credit-risk data products, the combination of native handling as a default and iterative random forest as a comparative robustness check offers a defensible baseline.

5.2. Limitations and Regulatory / Fairness Considerations

Several limitations temper the generalizability of these findings. The injected MNAR protocol, though calibrated to mimic self-censoring via a feature-wise two-tailed logistic gate, cannot capture all real-world non-random patterns; genuinely missing-not-at-random fields in production data may follow richer mechanisms tied to channel, device, or temporal drift. The complete-feature filter in §3.1 inevitably narrows each dataset to the subset of records for which the analysis features are observed, thereby introducing selection bias: in particular, native missingness rates above 50% in Lending Club delinquency-recency variables are excluded from the experiments because the corresponding columns are removed before injection. The three public datasets also postdate their originators' own internal cleaning, so part of the native missingness has already been resolved before public release. The matched-condition variant addresses one specific concern about S5---namely, that its lead might depend on a clean test split---but does not, by itself, certify that conclusions transfer to operational scoring data, where missingness mechanisms can vary across channels. A further limitation is that the native tree-handling baseline is not a pure imputation method: it is a missingness-aware modeling strategy that can use the presence of missingness as predictive structure. Therefore, its advantage over pre-imputation methods should be interpreted as evidence in favor of native missing-value handling in tree-based credit models, rather than as proof that native handling would dominate all imputation methods if identical missingness-indicator information were supplied to every strategy. Subgroup analysis is restricted to a credit-history-depth definition of thin-file status, and the operational proxy on Home Credit relies on application-table fields rather than on a true bureau-history-age field; Give Me Some Credit is omitted entirely from the subgroup analysis because its 11 features cannot support a comparable proxy. Any fairness assessment should therefore proceed through the relevant adverse-impact, reason-code, and model-risk review processes, with the imputation stage treated as one of several interlocking preprocessing decisions rather than as a stand-alone compliance control.

5.3. Conclusion and Directions

This paper has documented how the choice among five common imputation strategies interacts with missingness rate, missingness mechanism, feature type, and credit-history depth on three public consumer-credit datasets. The practical takeaway is that native tree-based missing-value handling and iterative random-forest imputation together form a resilient default pair: they track the performance of more elaborate methods at low missingness and pull clearly ahead at elevated rates, particularly under non-random patterns and on limited-credit-history applicants, and the lead is preserved under the matched-condition variant in which test inputs themselves carry injected missingness. Simple filling, while computationally appealing, incurs a measurable, mechanism-sensitive penalty that scales with the rate. Two directions warrant further empirical work. The preprocessing stage studied here could be extended to longitudinal settings in which missingness patterns themselves drift across quarters, stressing any imputer whose parameters are estimated on a static training snapshot. A complementary line of work would connect the choice of imputation strategy to the downstream adverse-action reason-code pipeline, quantifying how differently each strategy alters the ordering and content of reason codes delivered to declined applicants.

References

1. S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124--136, 2015.

2. R. Florez-Lopez, "Effects of missing data in credit risk scoring: A comparative analysis of methods to achieve robustness in the absence of sufficient data," *Journal of the Operational Research Society*, vol. 61, no. 3, pp. 486--501, 2010.
3. S. Van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in R," *Journal of Statistical Software*, vol. 45, no. 3, pp. 1--67, 2011.
4. D. J. Stekhoven and P. Bühlmann, "MissForest---Non-parametric missing value imputation for mixed-type data," *Bioinformatics*, vol. 28, no. 1, pp. 112--118, 2012.
5. D. Bertsimas, C. Pawlowski, and Y. D. Zhuo, "From predictive methods to missing data imputation: An optimization approach," *Journal of Machine Learning Research*, vol. 18, no. 196, pp. 1--39, 2018.
6. J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 5689--5698, PMLR, 2018.
7. P.-A. Mattei and J. Frellsen, "MIWAE: Deep generative modelling and imputation of incomplete data sets," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97, pp. 4413--4423, PMLR, 2019.
8. N. B. Ipsen, P.-A. Mattei, and J. Frellsen, "not-MIWAE: Deep generative modelling with missing not at random data," in *International Conference on Learning Representations (ICLR)*, 2021.
9. A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera, "Handling incomplete heterogeneous data using VAEs," *Pattern Recognition*, vol. 107, p. 107501, 2020.
10. B. Muzellec, J. Josse, C. Boyer, and M. Cuturi, "Missing data imputation using optimal transport," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 7130--7140, PMLR, 2020.
11. J. You, X. Ma, D. Y. Ding, M. J. Kochenderfer, and J. Leskovec, "Handling missing data with graph representation learning," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 19075--19087, 2020.
12. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785--794, ACM, 2016.
13. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146--3154, 2017.
14. M. Le Morvan, J. Josse, E. Scornet, and G. Varoquaux, "What's a good imputation to predict with missing values?," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 11530--11540, 2021.
15. D. Jarrett, B. C. Cebere, T. Liu, A. Curth, and M. van der Schaar, "HyperImpute: Generalized iterative imputation with automatic model selection," in *Proceedings of the 39th International Conference on Machine Learning*, vol. 162, pp. 9916--9937, PMLR, 2022.
16. LendingClub Corporation, *Lending Club loan data, 2007--2020Q3 (wordsforthewise consolidated release)* [Data set], Kaggle, Aug. 12, 2024. [Online]. Available: https://www.kaggle.com/datasets/wordsforthewise/lending-club
17. Home Credit Group, *Home Credit Default Risk competition data* [Data set], Kaggle, Aug. 12, 2024. [Online]. Available: https://www.kaggle.com/competitions/home-credit-default-risk/data
18. W. Cukierski, *Give Me Some Credit competition data* [Data set], Kaggle, Aug. 12, 2024. [Online]. Available: https://www.kaggle.com/competitions/GiveMeSomeCredit/data