

Article

Overlap Mitigation Versus Classifier Selection in Imbalanced Classification: A Dual-Baseline Analysis

Sami Kaya *  and Ramazan Ünlü 

Department of Industrial Engineering, Abdullah Gül University, Kayseri 38080, Türkiye;
ramazan.unlu@agu.edu.tr

* Correspondence: sami.kaya@agu.edu.tr

Abstract

In imbalanced binary classification, class overlap forces a familiar fork in the road: refine a preprocessing pipeline, or pivot to classifier selection. Which path dominates has remained empirically open. To address this question, a full factorial experiment was run across 58 KEEL datasets, 11 mitigation techniques, and 10 classifiers (yielding 6960 fitted models in total), with results read through a dual-baseline lens. Against the same classifier without mitigation, the techniques delivered a mean improvement of +4.13 points; gains concentrated on weak learners, with SMOTE and BorderlineSMOTE topping the rankings. Against an honest leave-one-dataset-out (LODO) baseline that selects the strongest standalone classifier from a held-out training pool, mean $\Delta F1$ fell to -4.64 points, and 43.2% of mitigated configurations matched or beat the baseline. Among controllable factors, a Type II ANOVA placed classifier identity at $\eta^2 = 0.157$ against mitigation technique at $\eta^2 = 0.010$, so the choice of algorithm carried roughly sixteen times the explanatory weight of the choice of preprocessing. Dataset characteristics dominated the overall variance at $\eta^2 = 0.620$. The honest LODO selector returned LightGBM on every one of the 58 datasets, suggesting that under typical practitioner conditions, a strong-classifier default is the lower-effort, lower-variance path, with no expected performance cost relative to a pipeline-development investment.

Keywords: algorithm selection; class overlap; classification; imbalanced learning; preprocessing; SMOTE

1. Introduction

The class overlap, as a matter of consensus, degrades classification performance [1,2]. The contested question is what should be done in response. Two approaches compete: preprocessing the contested feature-space regions away or selecting a classifier with sufficient internal capacity to absorb them. Preprocessing has been more in the spotlight of research to date [3], which motivates this study to investigate the asymmetry in the field.

In regions of the feature space occupied simultaneously by instances from competing classes, no amount of resampling recovers the separability that the features themselves cannot provide [2]. The existence of imbalance in overlapped data further amplifies the challenge, where minority instances, surrounded by opposite class samples, show elevated classification difficulty [4]. Thabtah et al. [5] demonstrated that a rise in the imbalance ratio degrades the classification with a positive acceleration rather than linearly.

The Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTE+ENN) runs in two stages [6]. Minority instances are generated by



Academic Editor: Douglas
O'Shaughnessy

Received: 10 June 2026

Revised: 2 July 2026

Accepted: 5 July 2026

Published: 9 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

interpolation first, and then synthetic instances within the majority area are discarded. Operating at a finer spatial resolution, Tomek Links removes cross-class nearest-neighbor pairs directly [7]. The Neighborhood Cleaning Rule (NCR) runs in an asymmetric way; it removes the majority instances failing the local vote, but keeps minority instances [8]. Subsequent variants added layer upon layer of refinement [9,10]. None of these methods has been tested against strong classifiers trained with unprocessed data.

That gap guided the design of the experimental pipeline in our study. Crossing 58 binary datasets (N1 overlap: 0.00–0.42) with 11 mitigation techniques and 10 classifiers—yielding 6960 experimental runs—permitted a full factorial comparison under two distinct performance baselines. The first compared each mitigated algorithm against its own untreated performance on the same dataset. Against the second, the strongest unmitigated classifier available on each dataset, the same mitigated configurations were evaluated independently.

Under the same-classifier comparison, mitigation produced a mean gain of 4.13 F1 percentage points, with the gains concentrated in weaker algorithms where the margin for improvement was widest. Against an honest LODO baseline that selects the strongest classifier from a held-out training pool, mitigated pipelines trailed by 4.64 F1 points on average, although 43.2% of configurations matched or beat the baseline. Mitigation improves the performance of weak classifiers, but a strong-classifier baseline still outperforms it on the same datasets.

Two readings emerge from the experiments, and they cut in opposite directions. Mitigation justifies itself on the same-classifier comparison: a +4.13-point mean gain for weak learners is real and survives the analysis. Set the same configurations against a validation-grounded classifier choice, however, and they fall 4.64 points behind that honest baseline, with 43.2% reaching or surpassing it. The Type II ANOVA confirms the asymmetry; among controllable factors, classifier identity carries roughly sixteen times the explanatory weight of the mitigation technique. Above both sits dataset identity, which accounts for close to three-fifths of total F1 variance and motivates a separate research direction on dataset-level difficulty estimation that we pursue in a companion work currently under review.

A reader might counter with the ceiling effect—perhaps no headroom for improvement existed in the first place. A direct test of this demanded stratifying by baseline difficulty. Even in the lowest stratum, mitigation still produced degradation; the honest LODO baseline F1 of 0.48 there leaves more than half of the metric range available. The headroom argument does not survive this test.

Parts of the present picture have been reported before. Blagus and Lusa [11] showed that SMOTE's benefits erode as dimensionality rises; in a 179-classifier benchmark, Fernández-Delgado et al. [12] found Random Forest variants near the top regardless of dataset characteristics. The baseline-comparison confound remained unexamined by both studies. While some papers endorse preprocessing [13], others find it ineffective [11]; though both report accurately, they take distinct reference points. Prior comparison studies at smaller scales [14,15] fell short of the statistical mass to identify this pattern.

The factorial infrastructure assembled here—6960 runs spanning the full classifier-by-technique grid—provides that mass and yields two separable contributions: the resolution of a baseline-comparison confound that has made the preprocessing literature appear self-contradictory, and the empirical rejection of the ceiling-effect defense across the complete difficulty spectrum, including datasets where F1 improvement room was demonstrably available. Section 2 reviews overlap measurement and the mitigation approaches included in the comparison; Sections 3 and 4 present the methodology and the dual-baseline results; Section 5 addresses the implications and limitations before Section 6 concludes.

2. Literature Review

2.1. Class Imbalance in Machine Learning

In fraud detection, equipment fault monitoring, and clinical screening alike, the target event is structurally rare—balanced class distributions are the exception rather than the norm in applied machine learning [16]. Classifiers trained to minimize aggregate error respond in a well-documented way: majority-class predictions incur little cost in overall accuracy even as recall of the minority class collapses [17,18]. The problem has been recognized for decades, and the responses to it have diverged into three broad families [19,20].

The first set of approaches shapes the training distribution by undersampling, oversampling, or synthetic data generation [19,20]. Among the available methods, SMOTE [13] achieved the broadest uptake: by interpolating between existing minority instances and their k -nearest neighbors along connecting line segments, it creates new training examples in the minority-class vicinity. It has been widely adapted and criticized. Blagus and Lusa [11] showed that the instances generated by interpolation fail to overcome the majority-class dominance in high-dimensional feature spaces. That limits the practical scope of the SMOTE technique. Within an overlapping domain, SMOTE may generate synthetic samples in that very region of overlap and increase the complexity, instead of reducing it, an overlooked failure mode.

The second family modifies the learning algorithm rather than the data. This family focuses on classification models within two major categories. Cost-sensitive approaches apply higher penalty values for minority class instances [21], while ensemble methods aggregate distinct classifiers and reach a consensus with a more robust structure [22]. Boosting algorithms carry this logic further by iteratively concentrating training effort on instances the ensemble currently misclassifies. The interaction between overlap and the aforementioned algorithmic strategies has not been systematically examined [23]; we address this gap in our study.

2.2. Class Overlap Phenomenon

Overlap and imbalance are frequently treated as a single compound difficulty, but they are distinct in nature. Class imbalance is an issue of distribution that can be managed through resampling approaches. On the other hand, class overlap is a separability complexity. When instances from opposing classes occupy the same region of the feature space [2], the available features cannot fully discriminate between them regardless of how training counts are adjusted. The sources of class overlap include domain ambiguity around the class boundaries, feature-related issues, noise, and labeling errors [24].

In controlled experiments, Denil and Trappenberg [25] traced classification difficulty to overlap rather than to imbalance. Prati et al. [26] confirmed that imbalance alone, in the absence of overlap, does not significantly degrade classifier performance. More recently, Carvalho et al. [27] and Santos et al. [28,29] reviewed how their joint effects amplify each other. Vuttipittayamongkol et al. [30] demonstrated that this combination is more damaging than either factor alone.

2.3. Overlap Measurement Techniques

Construction begins with a minimum spanning tree over the full training set with class labels withheld; any instance whose tree neighbor belongs to the opposite class is then flagged. N_1 is the count of flagged instances divided by the total, where zero indicates complete class separation and a value approaching 0.5 indicates heavy class intermixing. Lorena et al. [31] assembled practical empirical guidance for the use of overlap metrics. Complexity metrics carry information that distributional statistics miss, a point established

by Al Hosni and Starkey [32]. A counterweight comes from Eberlain et al. [33], whose results show that data preprocessing fails to consistently reduce classification complexity measures, thereby casting doubt on the rationale for overlap mitigation.

Beyond N1, the data complexity literature catalogs several alternative measures. F1 (Fisher’s discriminant ratio) [34] quantifies how well features separate classes through projected variance. N3 [35] uses leave-one-out k-NN error to estimate boundary complexity. T1 [35] measures the number of hyperspheres required to cover each class without overlap. Both the Napierała–Stefanowski typology [36] and LOF [37] sort minority instances into four local-difficulty categories: safe, borderline, rare, and outlier. Each offers a different angle; N1’s geometric clarity and its scalability to the dataset count here made it the appropriate fit.

2.4. Mitigation Strategies

Three broad families organize the mitigation strategies in use. Undersampling methods reduce the majority class through cleaning rules: ENN [38] removes instances misclassified by their k-NN neighborhood; NCR [8] extends ENN with an asymmetric rule that protects minority instances; Tomek Links [7] removes mutual nearest neighbors across classes. Oversampling methods inflate the minority class: Random Oversampling duplicates instances; SMOTE [13] interpolates synthetic instances along k-NN segments; BorderlineSMOTE [39] confines synthesis to borderline regions; ADASYN [40] weights synthesis by local difficulty; CBO [41] clusters before synthesis. Hybrid methods combine the two—SMOTE+Tomek and SMOTE+ENN first oversample and then clean [6]. Newer contributions [42–44] develop overlap-aware variants that still sit within the same three-family structure.

2.5. The Algorithm Selection Question

The mitigation literature is substantial; what it has not produced is a direct, large-scale comparison between preprocessing and algorithm selection. The algorithm selection problem itself was formalized by Rice [45] and subsequently extended with complexity-based meta-features by Rivolli et al. [46], yet the imbalanced learning community has engaged with that framework only peripherally.

There exist prior signals. Comparing several balancing techniques, Batista, Prati, and Monard [6] found that relative effectiveness varied with both the dataset and the classifier. That observation leads to our conclusion, even though testing this was not the intention behind their design. The big review by Fernández et al. [19] is probably the most-cited piece in the imbalanced learning space, and it is well organized, but every comparison in it pits one technique against another technique, never a technique against a strong classifier running on raw data.

Galar et al. [47] reviewed ensemble models in imbalanced learning and revealed that ensembles outperform preprocessing-enhanced weak classifiers, but they did not decompose the variance explained by each factor. At a narrower scale, Zhang and Mani [48] showed that plain random undersampling matched more elaborate resampling schemes when k-NN served as the base learner. The largest classifier comparison relevant to this literature was conducted by Fernández-Delgado et al. [12]: 179 algorithms evaluated across 121 datasets, with Random Forest variants ranked consistently near the top regardless of data characteristics. Whether those rankings are driven by preprocessing or classifier selection is not directly assessed. All three studies agree that classifier choice outweighs data preprocessing.

Recent studies support this pattern without comprehensive testing. Dar and Farooq [49] combined overlap-driven undersampling with a balanced Random Forest—a design that

implicitly concedes the ensemble's standalone value. D'souza, M, and Sarawagi [50] introduced a ternary labeling scheme for synthetic data generation in which instances are explicitly tagged as belonging to overlap zones; the approach improved both the fidelity of generated data and downstream classifier training. Each of these contributions approaches the same question from a different angle, yet none delivers a direct, large-scale comparison of preprocessing value against algorithm selection.

Two recent benchmark studies bring the evidence closest to that comparison. Haluška and Brabec [51] tested 16 preprocessing techniques against an AUTO-sklearn baseline across 23 datasets; the baseline—involving no preprocessing at all—placed in the top half of rankings on both PR AUC and ROC AUC. Moniz and Cerqueira [52] automated the preprocessing search across 469 configurations using meta-learning, but held the classifier fixed at Random Forest throughout. Whether varying the classifier would have rendered the entire 469-configuration search unnecessary was not examined.

The present work positions itself against three threads in the literature. Batista, Prati, and Monard [6] articulate the conceptual distinction between same-classifier and best-available evaluation but stop short of quantifying its empirical magnitude; the dual-baseline framework operationalizes that distinction across a 58-dataset, 11-technique, 10-classifier factorial. A 179-classifier benchmark from [12] established Random Forest as a near-top default, yet it did not cross classifier choice with preprocessing—the relative weight of the two levers consequently remained unmeasured, and the variance decomposition reported below fills exactly that gap. Ensemble-based approaches for imbalanced learning received a structural review in [47], although the cross-classifier $\times$ cross-technique grid required to test the structural claims at scale was not run; that grid is what the factorial design here provides.

3. Methodology

3.1. Research Design

Three research questions guide the experimental design, each targeting a distinct aspect of the preprocessing-versus-classifier-selection comparison. RQ1 asks which classifiers hold up best when classes bleed into each other, while RQ2 asks whether preprocessing really improves classification compared to unprocessed data. RQ3 asks an even harder question: Can a weak classifier with mitigation outperform a strong classifier without it?

A critical difference emerges between research questions 2 and 3. Most of the current literature stops at RQ2—and by that measure, preprocessing helps classification. What RQ3 introduces is a practical constraint that the earlier literature did not confront. Algorithm switching is cost-free in most cases; however, it has not been tested at this scale whether preprocessing is redundant.

All experiments were conducted on a MacBook Pro (Apple Inc., Cupertino, CA, USA) equipped with an Apple M2 chip and 8 GB of unified memory.

3.2. Dataset Selection

We used 58 datasets from the Knowledge Extraction based on Evolutionary Learning (KEEL) dataset repository [53]. KEEL is widely used in the imbalanced learning community because it provides effectively standardized datasets, which enable cross-study comparisons. Selection applied four criteria. First, we included only binary class datasets, since N1 measurement among multiple classes would add complexity to the problem. Second, the imbalance ratios ranged between 5:1 and 130:1, covering the moderate-to-severe range that arises routinely in applied settings. Third, instance counts ranged between 129 and 1484—this kept five-fold cross-validation statistically applicable while keeping the design computationally tractable. Fourth, each dataset had to carry explicit KEEL metadata anno-

tations for borderline and noisy instances, since the overlap quantification step depended on that annotation.

The resulting collection spans several domains: abalone and paw (ecology and synthetic patterns), glass (industrial quality control), cleveland (cardiology), yeast and ecoli (genomics), page-blocks (document imaging), and three synthetic benchmarks—subcl, clover, and paw. N1 ranged from 0.00 on shuttle, where the two classes are essentially non-overlapping, to 0.42 on a synthetic dataset with deliberately mixed boundaries. Across the 58 datasets, N1 averaged 0.21 with a median of 0.22. Table 1 partitions the collection by the honest LODO baseline F1 (Section 3.7) on each dataset.

Table 1. Dataset characteristics by baseline performance stratum.

Stratum	Count	Mean N1	Mean IR	Mean Size	Mean F1
Low (<0.70)	40	0.25	11.3	633	0.48
Medium (0.70–0.80)	7	0.13	14.4	624	0.75
Good (0.80–0.90)	3	0.10	11.8	499	0.86
High (>0.90)	8	0.09	11.0	691	0.94
Total	58	0.21	11.7	633	0.60

The full 58-dataset characteristics table—instance counts, feature counts, imbalance ratios, minority counts, and the N1 and N3 complexity metrics—is provided in Appendix A.

3.3. Overlap Measurement

Class overlap was measured using Ho and Basu’s N1 metric [35]. A minimum spanning tree (MST) is constructed across the pooled training data with class labels suppressed. Each MST edge is then inspected, and any instance with at least one neighbor of the opposite class is tagged as borderline. The proportion of such boundary-crossing instances over the total defines N1.

At $N1 = 0$, the two classes occupy disjoint regions with no shared boundary structure; at N1 approaching 0.5, interclass interleaving is severe. The choice of N1 over other measures like Fisher’s discriminant ratio and N3 is based on three reasons. The interpretation of N1 is geometric and direct; the metric equals the fraction of instances with an opposing-class neighbor, and it requires no normalization. MST computation scales at $O(n \log n)$, making it viable across all 58 datasets without preprocessing overhead. The metric also carries an established track record in the imbalanced learning literature, reducing the likelihood of reviewer objections to the measurement framework.

3.4. Mitigation Techniques

The evaluation covered eleven techniques drawn from three families: undersampling, oversampling, and hybrid combinations.

Undersampling methods: Tomek Links identifies pairs of opposite-class instances that are mutual nearest neighbors and removes them from the training set [7]. ENN applies a 3-nearest-neighbor vote around each instance and discards any instance whose vote misclassifies it [38]. Repeated Edited Nearest Neighbors (RENN) extends ENN by iterating the cleaning pass until no further deletions occur—capped at ten passes in this experiment, though most datasets converged earlier. NCR operates asymmetrically: a majority-class instance misclassified by its 3-NN neighborhood is dropped, whereas a minority-class instance is only removed when all k neighbors belong to the majority [8]; this asymmetry is designed to prevent the cleaning step from deepening the imbalance it is supposed to address.

Oversampling methods: Random Oversampling duplicates minority-class instances uniformly. SMOTE selects a minority instance at random, identifies one of its five nearest

minority-class neighbors, and synthesizes a new instance by linear interpolation along the connecting segment [13]. BorderlineSMOTE confines synthesis to minority instances whose five-nearest-neighbor set contains between two and four majority-class neighbors—the borderline zone, in other words—leaving core and noise instances untouched [39]. ADASYN concentrates synthetic generation in the hardest neighborhoods by weighting sampling density according to local classification difficulty [40]. Cluster-Based Oversampling (CBO) identifies natural groupings via k-means prior to synthesis and generates new instances within each cluster’s bounds [41].

Hybrid methods: SMOTE+Tomek generates synthetic instances via SMOTE and then applies a Tomek Links sweep to remove any ambiguous pairs created in the augmented data [6]. SMOTE+ENN substitutes ENN cleaning for the Tomek pass. Mitigation techniques are implemented through the library imbalanced-learn 0.13.0 [54]. We used default hyperparameters on purpose, to avoid bias between fine-tuned and raw models.

3.5. Classification Algorithms

We selected 10 classifiers to represent principal model families and keep the experimental design manageable. We grouped the classifiers into four families based on their learning mechanism. Linear models: Logistic Regression (L2 penalty, $C = 1.0$, balanced weights), and Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel, ($\gamma = \text{‘scale’}$, $C = 1.0$), were used as capacity baselines against which tree-based gains could be measured. Classification and Regression Trees (CART) was implemented for single-tree learning with balanced weights and without a depth cap. The ensemble extensions built on that same splitting logic: Random Forest aggregated 100 such trees under balanced class weights [55], while Extra Trees added randomized split thresholds on top of Random Forest’s random feature subsets, also with 100 trees. We ran 100 estimators for XGBoost with default parameters and histogram-based splitting for LightGBM with balanced class weights.

The remaining three models represent inductive bias. k-NN ($k = 5$, inverse-distance weighting) is instance-based; Naive Bayes assumes Gaussian class-conditional densities; and the MLP stacked two hidden layers (100 and 50 neurons) with ReLU activations, trained via Adam with early stopping enabled.

All models, except XGBoost (xgboost 3.1.2) and LightGBM (lightgbm 4.6.0), were implemented using scikit-learn 1.0.2 [56]. We used fixed seed 42 in every run for reproducibility purposes. We excluded bagging and stacking due to the model crashes across certain data–mitigation technique pairs.

3.6. Experimental Protocol

The same pipeline was applied to each configuration—all 6960 of them—within our experimental protocol. We excluded feature selection to preserve overlap profiles of the datasets. Five-fold stratified cross-validation was applied throughout, with fold assignments held constant via random seed 42. Standardization parameters were estimated once per dataset before the cross-validation split and the same transform was then applied to every fold. This kept the scaling step identical across the baseline and mitigation arms, so any leakage entered both arms symmetrically and could not bias the dual-baseline comparison.

The arithmetic: 58 datasets $\times$ 11 techniques $\times$ 10 classifiers = 6380 mitigated experiments; 580 unmitigated baselines (one per classifier–dataset combination); grand total, 6960.

3.7. Analysis Framework

Three $\Delta F1$ quantities organize the experiments below; each isolates a different practitioner question, and each has a precise operational definition.

The same-classifier baseline subtracts unmitigated from mitigated F1 within a fixed classifier C and dataset D : $\Delta F1_{\text{same}} = F1(C, T, D) - F1(C, \emptyset, D)$, with T denoting the mitigation technique under test. This is the quantity that asks whether a given technique helps a given model on a given problem. The honest LODO baseline answers a different question—what does a validation-grounded classifier choice yield, before any mitigation is considered. For a target dataset D , the mean baseline F1 of each candidate classifier was computed across the remaining 57 datasets; the classifier with the highest cross-dataset mean was then deployed on D , and its F1 recorded as $F1_{\text{honest}}(D)$. On all 58 LODO iterations, the selector returned LightGBM. $\Delta F1_{\text{honest}} = F1(C, T, D) - F1_{\text{honest}}(D)$ compares each mitigated configuration against this dataset-blind baseline. The oracle baseline records the per-dataset maximum: $F1_{\text{oracle}}(D) = \max$ over candidate classifiers of unmitigated F1 on D . Because it peeks at the target dataset it cannot be deployed in practice, and we therefore treat it as a sensitivity check on the honest baseline rather than as a primary comparator.

The honest LODO baseline is the primary comparator in the analyses that follow, for two reasons. First, it is implementable—every step (cross-dataset selection on training folds, single-classifier deployment on the target) is something a practitioner can actually do, in contrast to the oracle’s reliance on test-set information. Second, it answers the methodological question raised in reviews: when a researcher has the freedom to select a classifier rather than fix one, what does a defensible selection procedure return, and does mitigation improve upon it? The same-classifier baseline holds the classifier choice constant by construction and so cannot speak to this question; the oracle overstates what is deployably achievable, so we report it only as an upper bound.

We fit Type II ANOVA on F1 through the statsmodels OLS routine, with classifier, technique, dataset, and the classifier $\times$ technique interaction as effects. The overall mean $\Delta F1$ across the 6380 mitigation configurations is reported with paired t -tests against zero, and per-technique means use the same test stratified by technique. The significance threshold throughout is $\alpha = 0.05$. The Matthews correlation coefficient was logged alongside F1 for every configuration and is reported in Table 2; on this metric, the qualitative ordering reproduces the F1 result, with classifier identity dominating the mitigation technique under the same Type II decomposition.

Table 2. Baseline performance without mitigation, ranked by mean F1 across the 58 datasets. The Best-on-N column reports the number of datasets on which each classifier achieved the highest baseline F1 (column sums to 58). The Mean MCC column is included for metric robustness.

Rank	Classifier	Mean F1	Std Dev	Mean MCC	Best on N
1	LightGBM	0.599	0.227	0.559	16
2	Neural Network	0.592	0.226	0.566	13
3	Decision Tree	0.568	0.216	0.527	5
4	Extra Trees	0.563	0.248	0.547	2
5	XGBoost	0.560	0.235	0.531	3
6	SVM	0.554	0.187	0.520	7
7	k-NN	0.552	0.257	0.534	7
8	Random Forest	0.546	0.255	0.527	2
9	Logistic Reg.	0.372	0.212	0.260	2
10	Naive Bayes	0.206	0.294	0.198	1

4. Results

4.1. Baseline Classification Performance

Table 2 reports mean F1 across all 58 datasets for each classifier operating on raw, non-preprocessed data.

LightGBM achieved the highest baseline F1 on 16 of the 58 datasets and Neural Network on 13, with the two leaders taking half of the collection between them (29 datasets, 50.0%). The four tree ensembles examined in subsequent sections—Random Forest, Extra Trees, XGBoost, and LightGBM—together led on 23 datasets, while each of the other six classifiers also led on at least one dataset, an outcome consistent with strong dataset-by-classifier heterogeneity rather than a single uniformly dominant model. Performance spans were wide: LightGBM’s mean F1 of 0.599 sat 39.3 points above Naive Bayes (0.206) and 22.7 points above Logistic Regression (0.372). One methodological note belongs to the baseline column itself. The submitted version of this manuscript reported a Neural Network mean of 0.553 against the present 0.592, together with seventeen F1 = 0 cases; that bimodal behavior arose because early stopping, enabled in the original protocol, occasionally split the training fold into a validation subset containing no minority instances on the smallest highly imbalanced problems, after which the optimizer converged on a majority-class predictor. Disabling early stopping removed the failure mode—no Neural Network F1 = 0 case remains, and the Neural Network mean lifts to 0.592 with median 0.578 (interquartile range 0.407–0.775). Pooled across the ten classifiers, the mean baseline F1 was 0.511, a figure better read as a statement about dataset difficulty than about classifier capability. The four tree ensembles (Random Forest, Extra Trees, XGBoost, LightGBM) anchor the analyses that follow as the operationally relevant strong-classifier set; the Neural Network, although now competitive on the F1 mean, is not selected by the honest LODO procedure on any of the 58 datasets and carries the highest fit cost and variance among the ten (SD = 0.226, mean fit time 0.736 s), so it is treated as a within-pool comparator rather than as a deployment default.

4.2. Same-Classifier Perspective: Mitigation Helps

Table 3 summarizes mean $\Delta F1$ for each technique when performance is measured against the same classifier run without any preprocessing.

Table 3. Same-classifier perspective—mean $\Delta F1$ by technique.

Technique	$\Delta F1$ (%)	N	Success
SMOTE	+6.36	580	68.4%
SMOTE+Tomek	+6.37	580	67.8%
BorderlineSMOTE	+6.12	580	65.2%
CBO	+5.85	580	65.2%
ADASYN	+5.65	580	63.8%
SMOTE+ENN	+4.69	580	58.8%
RandomOverSampler	+4.28	580	56.4%
ENN	+2.04	580	47.9%
RENN	+1.68	580	45.7%
NCR	+1.79	580	47.9%
Tomek Links	+0.58	580	39.7%

The 6380 mitigated configurations produced a mean $\Delta F1$ of +4.13 points ($\sigma = 11.96$; paired t -test against zero, $p < 0.001$), with 57.0% of them improving on their unmitigated counterparts. The split between oversampling and undersampling is sharp: SMOTE and SMOTE+Tomek co-lead at +6.36 and +6.37 points (68.4% and 67.8% success rates); Tomek Links trails at +0.58 points (39.7%). Two classifier groups clarify the mechanism. On the weak-learner side (Naive Bayes, Logistic Regression, SVM), mitigation delivered a same-classifier mean of +5.82 percentage points; on the tree-ensemble side (Random Forest, Extra Trees, XGBoost, LightGBM), the same-classifier mean was +4.04 points—the two groups sit at comparable magnitudes on this comparison. The Neural Network is not grouped

with either set because after the protocol correction described in Section 4.1 its baseline F1 sits at 0.592, above the weak-learner range and within the tree-ensemble band. The two oversampling hybrids, SMOTE+Tomek and SMOTE+ENN, produced results bracketed by their component methods.

4.3. Best-Available Perspective: Mitigation Falls Short

Pooled across the 6380 mitigation configurations, the mean $\Delta F1$ against the honest LODO baseline was -4.64 points ($\sigma = 16.90$), with the configuration matching or improving on the baseline in 43.2% of cases (Table 4). Against the oracle baseline the gap widened to -11.32 points ($\sigma = 15.40$, success rate 18.0%). Both losses exceed the $+4.13$ -point improvement against the same-classifier baseline in absolute magnitude, indicating that the same configurations that help a fixed weak learner fail to recover the gap to a classifier-aware default.

Table 4. Mean $\Delta F1$ (%) and success rate (%) across three baselines.

Technique	n	vs. Same-Classifier	vs. Honest LODO	vs. Oracle
SMOTE	580	+6.36 (68.4)	−2.41 (49.8)	−9.09 (24.1)
SMOTE+Tomek	580	+6.37 (67.8)	−2.40 (50.2)	−9.08 (23.3)
BorderlineSMOTE	580	+6.12 (65.2)	−2.65 (48.3)	−9.33 (19.7)
CBO	580	+5.85 (65.2)	−2.92 (46.0)	−9.60 (22.1)
ADASYN	580	+5.65 (63.8)	−3.12 (47.8)	−9.80 (22.9)
SMOTE+ENN	580	+4.69 (58.8)	−4.08 (43.3)	−10.76 (17.6)
RandomOverSampler	580	+4.28 (56.4)	−4.49 (36.0)	−11.17 (10.2)
ENN	580	+2.04 (47.9)	−6.73 (42.6)	−13.41 (18.1)
RENN	580	+1.68 (45.7)	−7.09 (40.3)	−13.77 (17.9)
NCR	580	+1.79 (47.9)	−6.98 (39.0)	−13.66 (14.7)
Tomek Links	580	+0.58 (39.7)	−8.19 (32.2)	−14.87 (7.6)

Mitigation techniques separate into two clusters under the honest LODO baseline. The seven oversampling and hybrid methods (SMOTE, SMOTE+Tomek, BorderlineSMOTE, CBO, ADASYN, SMOTE+ENN, RandomOverSampler) sit between -2.40 and -4.49 points, while the four undersampling methods (ENN, RENN, NCR, Tomek Links) sit between -6.73 and -8.19 points. The clusters overlap less at the extremes than at the boundaries: The best oversampling technique (SMOTE+Tomek at -2.40) sits roughly 3.4 times closer to zero than the worst undersampling technique (Tomek Links at -8.19), while the cluster-boundary gap (-4.49 for RandomOverSampler versus -6.73 for ENN) is closer to 1.5-fold. The within-cluster ordering matches the same-classifier ranking: SMOTE+Tomek and SMOTE co-lead at -2.40 and -2.41 , and Tomek Links trails at -8.19 . Success rates follow the same pattern, with the strongest technique reaching 50.2% of configurations and the weakest 32.2%.

The negative-mean pattern under strong-classifier baselines is robust to the choice of comparator. Across the four baselines tested—same-classifier, honest LODO, Random-Forest-only, and oracle—the same-classifier reading is positive ($+4.13$, success 57.0%) and the Random-Forest-only reading is marginally positive ($+0.67$, success 60.6%), while the honest LODO (-4.64 , success 43.2%) and oracle (-11.32 , success 18.0%) readings are negative (Figure 1). That the Random-Forest-only column sits so close to zero—mitigation neither systematically helps nor hurts when the comparator is a single strong ensemble—reproduces the headline qualitative finding without committing to LODO-style selection. What this implies is that classifier strength drives the result, not the specific selection rule; the honest LODO value falls in the negative direction only because LightGBM is a stronger comparator than Random Forest on this collection.

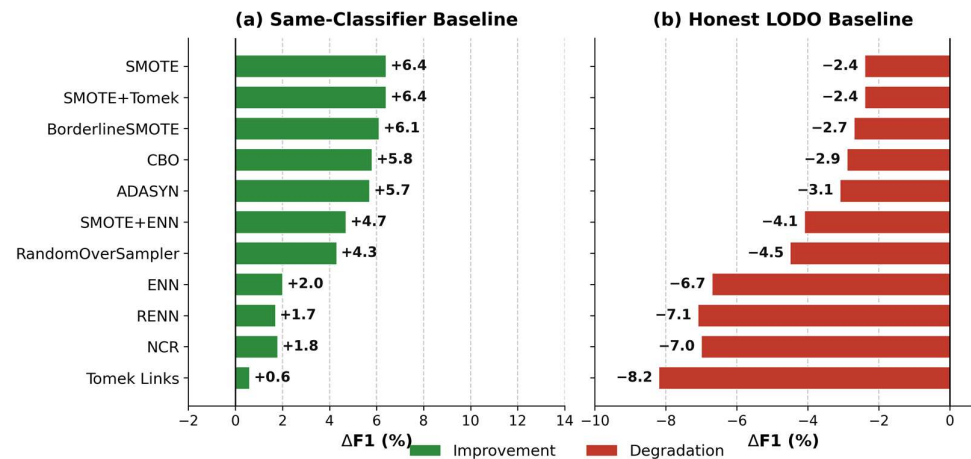


Figure 1. The two-baseline paradox. (a): under the same-classifier baseline, mitigation produces a mean $\Delta F1$ of +4.13 points. (b): under the honest LODO baseline (single strongest classifier selected from a held-out training pool), the same configurations register a mean $\Delta F1$ of −4.64 points. The same data yields opposing conclusions depending on the comparator.

4.4. Resolving the Paradox

The two opposing signals trace to a single structural asymmetry that survives the protocol correction, though it now appears at smaller magnitudes than the submitted analysis reported. On the weak-learner side (Naive Bayes, Logistic Regression, SVM), mitigation registered a same-classifier $\Delta F1$ of +5.82 points and an honest-baseline $\Delta F1$ of −16.34 points. The gain from mitigation on the same-classifier comparison was real, but the group remained roughly sixteen points below the strong-classifier ceiling on the honest baseline. On the tree-ensemble side (Random Forest, Extra Trees, XGBoost, LightGBM), the same-classifier $\Delta F1$ was +4.04 points and the honest-baseline $\Delta F1$ was +0.85 points—a near-zero honest mean that decomposes into 56.2% of configurations improving on the baseline and 30.2% landing within ± 2 points either side. For the tree-ensembles, mitigation is a near-zero-expectation move with non-trivial variance in both directions rather than a uniformly neutral or beneficial intervention. The Neural Network row is treated separately in this analysis for the reasons documented in Section 4.1: its post-fix same-classifier $\Delta F1$ sits at +3.04 and its honest-baseline $\Delta F1$ at +2.39, values that place it above the weak-learner cluster and inside the tree-ensemble band on the honest comparison.

In the majority of experiments, a strong classifier with no mitigation outperformed a weak classifier paired with SMOTE; that single observation, taken alone, is enough to anchor the practical reading. Mitigation took weak classifiers part of the distance to the ensemble ceiling without taking them all the way. The asymmetric tier-level pattern is what reconciles the two perspectives: the same-classifier baseline rewards the partial movement, while the honest LODO baseline measures the remaining gap.

4.5. Variance Decomposition

F1 variance was decomposed with a Type II ANOVA fit through the statsmodels 0.14.2 OLS routine, entering classifier, technique, dataset, and the classifier $\times$ technique interaction as effects. Table 5 reports the η^2 breakdown and Figure 2 visualizes the corresponding proportions of total variance.

Dataset characteristics took the largest share at $\eta^2 = 0.620$; problem identity alone accounts for close to three-fifths of F1 variance, before any methodological choice enters. Classifier identity followed at $\eta^2 = 0.157$, the heaviest weight among factors a practitioner can directly control. The classifier $\times$ technique interaction came in third at $\eta^2 = 0.026$, larger than the technique main effect and therefore inconsistent with a model in which mitigation

operates uniformly across classifiers, though small enough that the additive component dominates. Mitigation technique placed last at $\eta^2 = 0.010$, the smallest of the four shares.

Table 5. Variance decomposition (Type II ANOVA, full classifier pool, with interaction).

Factor	df	η^2	<i>p</i>
Dataset	57	0.620	<0.001
Classifier	9	0.157	<0.001
Classifier $\times$ Technique	99	0.026	<0.001
Technique	11	0.010	<0.001
Residual	6783	0.187	—

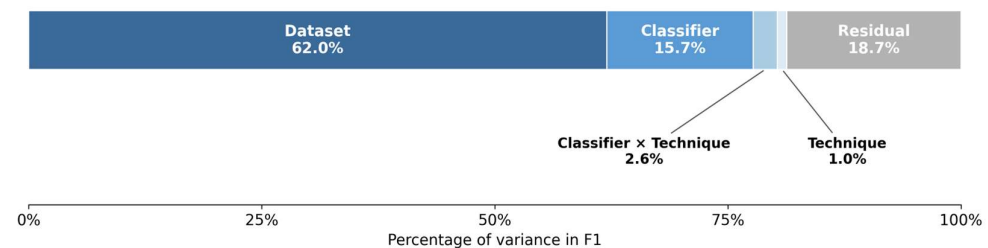


Figure 2. Variance decomposition (Type II ANOVA, full pool, model with classifier $\times$ technique interaction). Dataset characteristics dominate the decomposition; classifier identity explains approximately sixteen times more variance than mitigation technique among methodological choices.

Expressed as a ratio, classifier identity explains roughly sixteen times the F1 variance that mitigation technique explains among controllable factors. The qualitative ordering (dataset > classifier > technique) holds across all three pool definitions reported in Table 6, and the ratio quoted here is the more conservative of those available; the same ordering reappears on AUC, where classifier identity explains $\eta^2 = 0.296$ of the variance against $\eta^2 = 0.001$ for the technique, which leaves the AUC-based technique share too small to support a stable ratio statement. We therefore report only the η^2 values on that metric. That the dominant single factor is the dataset itself, rather than any methodological lever, motivates a separate research direction on dataset-level difficulty estimation, which we pursue in a companion work currently under review.

Table 6. Variance decomposition under classifier-pool removal (main-effects model).

Classifier Pool	n	η^2 Classifier	η^2 Technique	η^2 Dataset	η^2 Residual	Ratio
Full (10 classifiers)	6960	0.157	0.010	0.620	0.212	16.2:1
Neural Network excluded (9)	6264	0.160	0.011	0.607	0.222	14.7:1
NN and Naive Bayes excluded (8)	5568	0.113	0.004	0.722	0.162	32.1:1

The picture under pool removal looks different after the protocol correction described in Section 4.1. Dropping Neural Network from the classifier pool leaves the classifier share essentially flat— η^2 moves from 0.157 to 0.160—whereas dropping both Neural Network and Naive Bayes reduces it to $\eta^2 = 0.113$. The classifier-to-technique ratio is no longer monotonic in pool size: 16.2:1 on the full pool, 14.7:1 with Neural Network excluded, 32.1:1 with both excluded. Full and NN-excluded pools now sit close together because the corrected Neural Network no longer carries the kind of dispersion that previously inflated the full-pool ratio; the larger jump appears only when Naive Bayes—the one genuinely weak probabilistic learner remaining in the comparison—is also removed, at which point dataset characteristics absorb most of the redistributed variance and rise from $\eta^2 = 0.620$ to $\eta^2 = 0.722$. Across all three configurations the qualitative ordering (dataset > classifier > technique) holds.

The interaction share, while small, is not distributed evenly across classifier rows. Among the four tree-ensembles, per-cell residuals stay within roughly ± 6 points of the additive prediction; the rank ordering among them shifts modestly from technique to technique, but the four sit so close on baseline F1—between 0.546 and 0.599—that any rank churn is practically immaterial for an operator who will adopt one of the four. The residual interaction signal concentrates instead in the rows of the genuinely weak learners, most visibly Naive Bayes, whose response to oversampling versus undersampling differs by roughly twenty-four F1 points and supplies most of the interaction η^2 that survives the protocol correction. The cell-level pattern is examined in the next section.

4.6. Classifier $\times$ Technique Interaction

A type II ANOVA placed the classifier $\times$ technique interaction third among the four components, at $\eta^2 = 0.026$ ($F = 9.46$, $p < 0.001$). The interaction is therefore neither dominant—dataset and classifier carry far more weight—nor structurally absent: at roughly two and a half times the technique main effect, it rules out a model in which mitigation operates uniformly across classifiers, but the additive components describe most of the F1 surface, and the interaction adds a correction of modest magnitude on top of them.

Most of the surviving interaction signal sits on a single classifier row: Naive Bayes. Its largest positive cell residuals come from oversampling—Random Oversampler and CBO at +9.8 points, and ADASYN and BorderlineSMOTE at +8.9—and its largest negative residuals from undersampling—NCR at −14.0, ENN at −13.4, Tomek Links at −12.6, RENN at −11.9. The spread across techniques on this one row is roughly twenty-four F1 points, and it is the dominant contributor to the $\eta^2 = 0.026$ interaction share. Logistic Regression carries a much milder version of the same pattern (residual range [−2.4, +3.6]).

The submitted version of this analysis attributed the bulk of the interaction signal to the Neural Network, with per-cell residuals spanning roughly forty F1 points across techniques. That pattern turned out to be largely an artifact of the early-stopping behavior described in Section 4.1: under the original protocol, the optimizer collapsed to a majority-class predictor on a subset of small, highly imbalanced datasets whenever the validation split happened to contain no minority instances, which left the Neural Network's cell residuals dominated by oversampling-versus-undersampling differences in whether the collapse occurred at all rather than by genuine learning behavior. With early stopping disabled, the Neural Network cell residuals now sit within [−2.3, +2.0] across all eleven techniques—well inside the strong-classifier band—and the interaction signal that remains is carried by Naive Bayes, the one weak probabilistic learner that did not suffer from the protocol issue. The qualitative reading of the interaction (oversampling lifts weak probabilistic learners; undersampling starves them) is therefore unchanged from the submitted version; only the η^2 magnitude and the carrying classifier have shifted.

The four tree ensembles behave differently. For Random Forest, Extra Trees, XGBoost, and LightGBM, per-cell interaction residuals stay within [−5.8, +2.9] points across all eleven techniques—wider than in the submitted version but still narrow relative to the Naive Bayes row. The four also sit very close to one another on the raw baseline F1 (0.546–0.599, a 5.3-point span), and the rank ordering among them shifts modestly from technique to technique; a stronger rank-stability claim, of the sort the submitted version made through pairwise Kendall correlation, is not supported by the post-correction data. The practical point is unaffected, however: any operator who has settled on the tree-ensemble set as the deployment pool can expect a comparable F1 on the chosen target regardless of which member of the set is picked, and the choice among them can be made on cost, latency, or interpretability grounds rather than on expected accuracy.

The same asymmetry supplies a mechanistic reading of the dual-baseline result. Over-sampling lifts the genuinely weak learners substantially—Naive Bayes registers a same-classifier $\Delta F1$ of +16.92 points and Logistic Regression +0.20 points (its lower number reflects that L2-regularized logistic regression with class weights is already partially defended against imbalance)—large enough on the Naive Bayes row to make the aggregate +4.13-point same-classifier improvement plausible (Figure 3). The lift, however, terminates well short of the strong-classifier ceiling: against the honest LODO baseline (LightGBM on all 58 datasets), the same Naive Bayes configurations sit at −22.36 points on average, and Logistic Regression at −22.53. No mitigation technique converts a weak learner into a strong one, and the honest LODO baseline measures exactly the gap that oversampling does not close. That gap is why the pooled mean $\Delta F1$ against the honest baseline is −4.64 points rather than the +4.13 the same-classifier comparison reports.

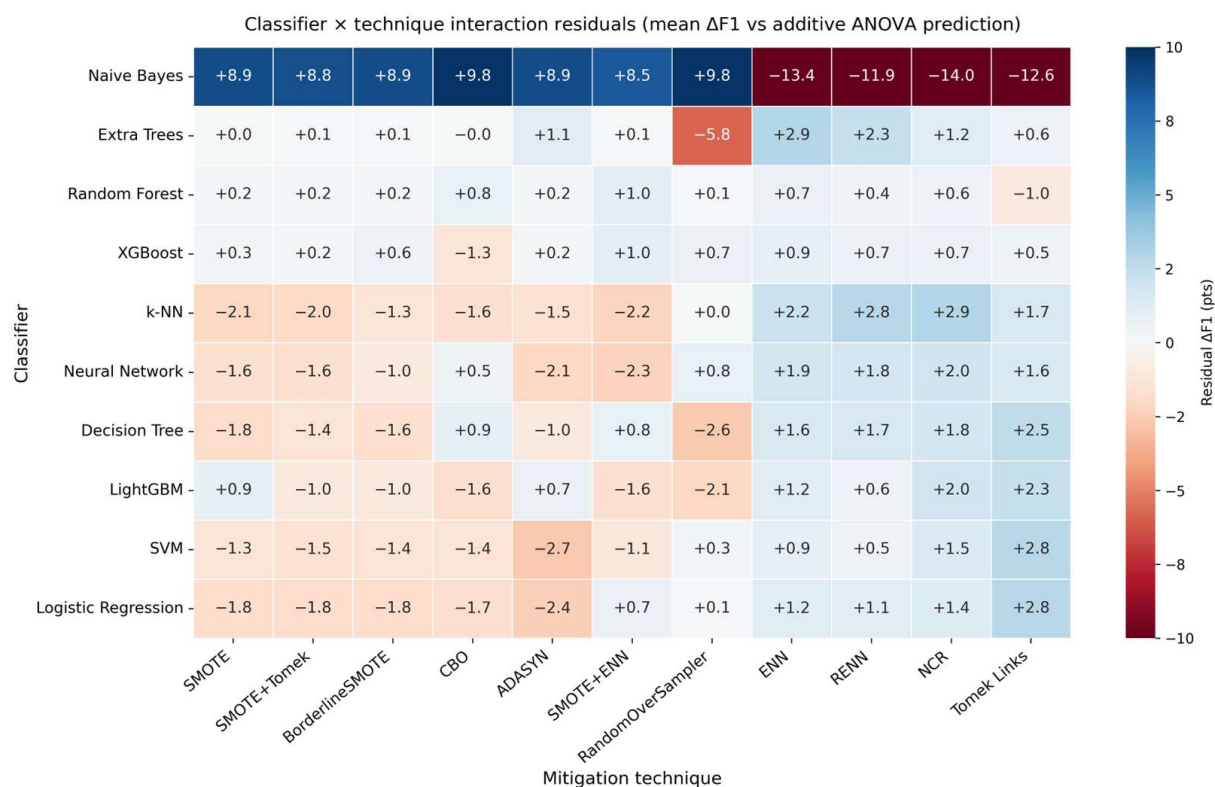


Figure 3. Classifier × technique interaction heatmap. Cell values show mean $\Delta F1$ residual relative to the additive ANOVA prediction. Naive Bayes carries the strongest residual signal—large positive values under oversampling (CBO and Random Oversampler at +9.8, ADASYN and BorderlineSMOTE at +8.9) and large negative values under undersampling (NCR at −14.0, ENN at −13.4, Tomek Links at −12.6, RENN at −11.9). After the early-stopping correction described in Section 4.1, Neural Network cell residuals sit within [−2.3, +2.0] across all eleven techniques. Among the four tree-ensembles (Random Forest, Extra Trees, XGBoost, LightGBM) cell residuals stay within [−5.8, +2.9] points.

4.7. Ruling out Ceiling Effects

A reasonable counterargument comes as ceiling effect, that ensemble models produce a score that is the maximum attainable F1 and leaves no room for preprocessing to show an improvement. We used stratified analysis to test this directly. Table 7 partitions the 58 datasets into four strata by honest LODO baseline F1 and reports mean $\Delta F1$ within each.

Each of the four strata returned a negative mean $\Delta F1$, significant at $p < 0.001$ (Figure 4). The crux is the low stratum, where the baseline F1 sits at 0.48 and 51.6% of the F1 scale remains available; nevertheless, the mean $\Delta F1$ is still −2.03% in this wide-open regime. Ceiling effects therefore cannot explain the degradation observed under the honest LODO

baseline, since more than half the metric range is empty in the stratum where the baseline is weakest. The degradation was actually more severe on easier datasets: the high stratum (baseline F1 > 0.90) returned -10.53% , compared with -2.03% in the low stratum. With less headroom available, mitigation perturbed the data distribution in ways that the strong-classifier baseline was already handling internally, and the perturbation cost showed up most clearly where that baseline was otherwise performing well.

Table 7. Stratified analysis by baseline performance.

Stratum	N	Baseline F1	Room (%)	Mean Δ F1	<i>p</i> -Value
Low (<0.70)	40	0.48	51.6%	-2.03%	<0.001
Medium (0.70–0.80)	7	0.75	24.6%	-9.04%	<0.001
Good (0.80–0.90)	3	0.86	14.4%	-13.44%	<0.001
High (>0.90)	8	0.94	5.7%	-10.53%	<0.001

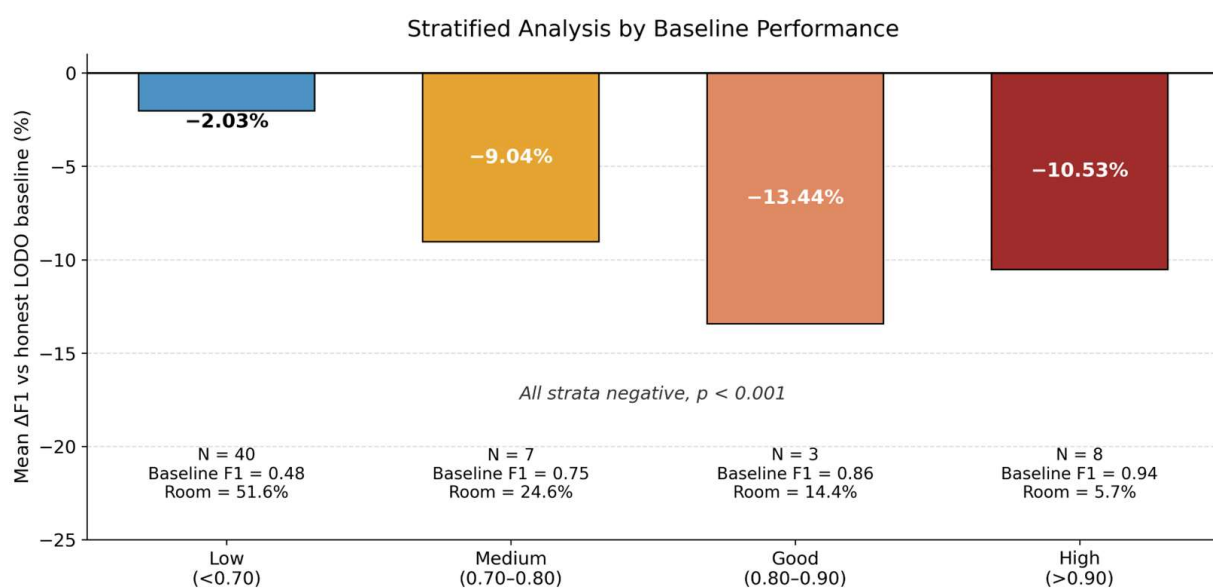


Figure 4. Stratified analysis by baseline performance. All four strata return negative mean Δ F1 against the honest LODO baseline. The low stratum (baseline F1 = 0.48, 51.6% of the metric range available) returns -2.03% , refuting the ceiling-effect explanation.

4.8. Technique Effectiveness by Stratum

Oversampling techniques cluster near zero in the low stratum (Figure 5). The three SMOTE-family entries in that column—SMOTETomek, SMOTE, and BorderlineSMOTE—sit at +0.37, +0.36, and +0.16 points, respectively, and they are the only cells in the 44-cell grid to exceed zero at all; even those gains are of a magnitude that would be absorbed by the run-to-run variance of the individual F1 measurements. That corner is where mitigation comes closest to competitive: the SMOTE-family techniques applied to difficult datasets neither help nor hurt the honest LODO baseline in aggregate. Every other cell carries a negative sign—the SMOTE family reaches roughly -7 points in the medium stratum and between -9 and -12 in the good and high strata, and the four undersampling methods (ENN, NCR, RENN, Tomek Links) fall further: roughly -4 to -6 points in the low stratum and between roughly -11 and -16 in the remaining three. Deleting instances from already-small minority classes removed more signal than boundary-cleaning recovered. The mitigation ceiling therefore sits at zero, is reached only in one corner of the grid (SMOTE-family $\times$ low stratum), and is not exceeded anywhere by more than four-tenths of an F1 point.

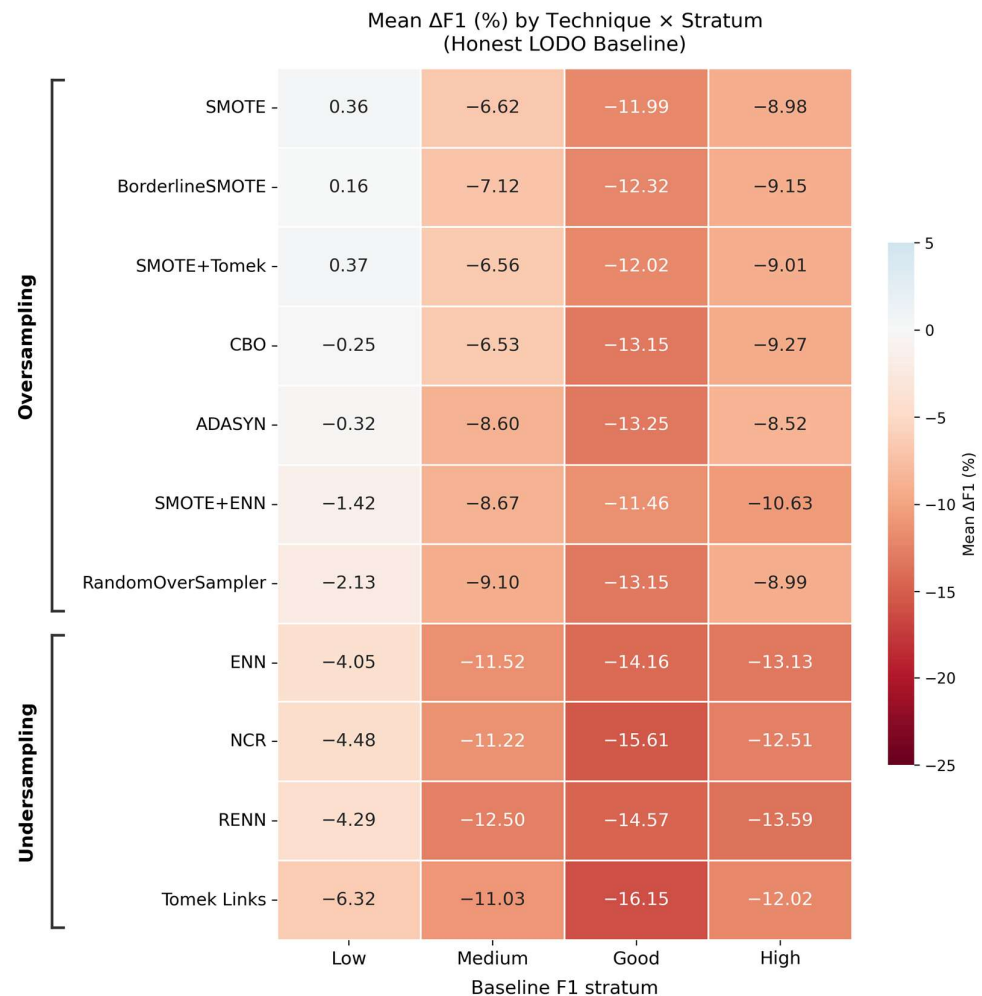


Figure 5. Technique effectiveness heatmap by baseline stratum. Three of the 44 cells sit marginally above zero (SMOTETomek, SMOTE, and BorderlineSMOTE in the low stratum at +0.37, +0.36, and +0.16 points respectively); every other cell returns a negative mean $\Delta F1$ against the honest LODO baseline. Oversampling techniques cluster nearest zero; undersampling techniques carry the largest losses, particularly in strata above $F1 = 0.70$.

5. Discussion

5.1. Reconciling Contradictory Findings

The published literature falls into two camps, and both report accurately. Chawla et al. [13] conclude that SMOTE is indispensable for decision trees; Fernández-Delgado et al. [12] conclude that Random Forest performs strongly without any preprocessing assistance. The discrepancy is not methodological error on either side; it is a baseline problem. The two camps have been measuring different quantities all along.

For Camp A, the question is whether preprocessing improves a classifier compared with itself. Camp B asks the bigger question: whether any preprocessed configuration can match a strong-classifier baseline on raw data—and that is what we operationalize here through the honest LODO baseline. Camp A finds gains; Camp B finds parity at best, and more often, losses. Neither reading is wrong; they measure different quantities.

The practical relevance of a finding turns on whether algorithm selection is available to the practitioner. Where regulatory or operational constraints lock in the algorithm—say, a credit-scoring deployment that requires logistic regression for auditability—Camp A's guidance applies directly: preprocess, and SMOTE serves as a defensible default. When the

practitioner has the freedom to select an algorithm, our findings suggest that preprocessing can be skipped in favor of the best available classifier.

5.2. Why Ensemble Methods Handle Overlap

Random Forest owes its robustness to the bootstrap aggregation mechanism [57]. Across hundreds of trees, each trained on a distinct random subsample with a distinct random feature subset, some trees will draw feature combinations that resolve the class boundary even in overlap-dense regions; others will not, but their errors cancel during aggregation. Overlap regions are precisely where individual trees disagree most—and the averaging operation is most effective exactly there. Bagging makes ensembles tolerant to class boundary ambiguity.

XGBoost arrives at the same outcome through a different mechanism [58]. Each successive tree in the boosting sequence is trained on a reweighted version of the data that upweights instances misclassified in the previous round. The boosting sequence progression outweighs the instances close to the decision boundary. The boosting loop therefore redirects successive trees toward exactly the overlap zones that preprocessing pipelines try to clean externally. The key point is not whether overlap management lies in the data pipeline or the classifier, but its presence somewhere.

D'souza et al. [50] developed a synthesis approach that explicitly tags instances by their distances to the class boundaries before generating synthetic instances from the minority class to improve the data quality for minority synthesis. The deeper implication of that work aligns with what our variance decomposition shows: the structural advantage of ensembles over preprocessing is not about any single dataset or technique but about where in the pipeline overlap handling occurs. Modern ensembles carry that logic internally. Bolting external resampling on top of an internally robust classifier adds little—and, as Table 8 shows, sometimes actively interferes.

Table 8. Mean $\Delta F1$ (%) by technique and stratum (honest LODO baseline).

Technique	Low	Medium	Good	High
SMOTE	+0.36	−6.62	−11.99	−8.98
BorderlineSMOTE	+0.16	−7.12	−12.32	−9.15
SMOTE+Tomek	+0.37	−6.56	−12.02	−9.01
CBO	−0.25	−6.53	−13.15	−9.27
ADASYN	−0.32	−8.60	−13.25	−8.52
SMOTE+ENN	−1.42	−8.67	−11.46	−10.63
RandomOverSampler	−2.13	−9.10	−13.15	−8.99
ENN	−4.05	−11.52	−14.16	−13.13
NCR	−4.48	−11.22	−15.61	−12.51
RENN	−4.29	−12.50	−14.57	−13.59
Tomek Links	−6.32	−11.03	−16.15	−12.02

5.3. Practical Recommendations

For practitioners who can freely select among algorithms: fit a strong classifier—Random Forest, Extra Trees, XGBoost, or LightGBM—on the raw, unmodified data first. The honest LODO selector returned LightGBM on every one of the 58 datasets in this collection, making it a validation-grounded default. The ensemble methods outperformed mitigated weak classifiers in roughly 57% of the configurations tested (mean $\Delta F1$ of −4.64 points against the honest LODO baseline). The time required to build, validate, and tune a preprocessing pipeline would, for the majority of problems in this collection, have been better spent on algorithm selection and hyperparameter tuning. A descriptive pattern visible in the aggregate—mean $\Delta F1$ more negative for datasets with baseline F1 above approximately 0.70—does not function

as a deployable decision rule: an out-of-sample test of an $F1 > 0.70$ cutoff returned 39.6% accuracy on held-out data, below the chance reference for the binary apply-or-skip decision.

Oversampling is the less risky preprocessing option whenever ensemble methods are off the table—for example, when interpretability is required, when latency budgets rule out large forests, or when edge deployment imposes tight memory limits. SMOTE and BorderlineSMOTE raised the underlying classifier's $F1$ in 65–68% of same-classifier comparisons.

Undersampling is a poor first-line choice for most binary imbalanced problems. ENN, RENN, and NCR produced modest mean gains paired with high variance; Tomek Links was nearly inert at +0.58 points under same-classifier evaluation and severely negative at −8.19 points under the honest LODO baseline. Removing instances from an already minority-sparse dataset tends to eliminate exactly the signal the classifier needed most.

A final caveat applies across all recommendations: these figures are aggregate summaries across 58 datasets. Particular datasets routinely diverge from the average pattern. Cross-validated comparison on the specific problem at hand remains the appropriate basis for any final decision.

Compute cost is not the binding constraint for the recommendation against routine mitigation. Mean preprocessing time across the eleven techniques sits below 12 ms per dataset (Table 9), and even the slowest classifier in the pool—Neural Network at 0.74 s mean fit time—completes a five-fold pass in roughly four seconds. The issue is not that mitigation is expensive, but that the $F1$ it returns on average does not justify the validation, hyperparameter search, and pipeline-engineering work needed to deploy it responsibly, given that an unmitigated tree ensemble already reaches a higher ceiling on the same data.

Table 9. Mean fit time per classifier (panel a) and mean preprocessing time per technique (panel b), in seconds, averaged across the 6960 configurations on the experimental hardware described in Section 3.1. Times are illustrative and depend on hardware; the ordering is robust and the preprocessing-time column stays below 12 ms across all techniques.

(a) Mean Fit Time per Classifier (s)	
Classifier	Mean Fit Time (s)
Neural Network	0.736
LightGBM	0.213
XGBoost	0.081
Random Forest	0.057
Extra Trees	0.041
SVM	0.030
Logistic Regression	0.001
Decision Tree	0.001
Naive Bayes	0.000
k-NN	0.000
(b) Mean Preprocessing Time per Technique (s)	
Technique	Mean Preprocessing Time (s)
CBO	0.012
RENN	0.005
SMOTE+ENN	0.004
SMOTE+Tomek	0.003
NCR	0.003
ENN	0.001
Tomek Links	0.001
BorderlineSMOTE	0.001
ADASYN	0.001
SMOTE	0.001
RandomOverSampler	0.000

5.4. Limitations

Several scope choices limit how far these findings can be generalized, and each deserves to be stated openly.

Every classifier and mitigation technique ran at library defaults rather than under per-dataset hyperparameter tuning. This design choice mirrors how practitioners most often operate, and it removes the confound of tuning-budget asymmetry from the comparison, but it leaves open whether a tuned mitigation pipeline could close the gap to a tuned ensemble baseline. A follow-up study currently in preparation addresses precisely that question by tuning both arms under a matched compute budget.

The experiments cover binary classification only. Multi-class settings bring methodological decisions the present framework was not designed to handle—one-vs-rest versus one-vs-one decomposition, macro versus micro averaging, and per-class imbalance ratios all become relevant. The dataset collection is also bounded in three other respects that determine the operational scope of the recommendations: instance counts span $n \in [108, 1829]$, imbalance ratios $IR \in [5.0, 41.4]$, and overlap $N1 \in [0.00, 0.42]$. Recommendations in Section 5.3 should be read as applying to small-to-medium tabular binary problems matching the KEEL profile; large-scale tabular benchmarks with millions of rows, extreme-imbalance settings beyond $IR \approx 50$, and overlap regimes above $N1 = 0.5$ were not represented, and conclusions should not be extrapolated into those regimes.

F1 served as the primary performance metric throughout. A parallel decomposition on AUC reproduces the qualitative finding more sharply—classifier identity explains $\eta^2 = 0.296$ of AUC variance against $\eta^2 = 0.001$ for the mitigation technique—but the technique share on AUC is so small that any ratio statement on that metric would be numerically unstable, which is why we treat the AUC decomposition as a directional confirmation and report the F1-based ratio (approximately 16:1) as the headline. Out of scope for the preprocessing-versus-classifier-selection question, and therefore not evaluated, were feature engineering, cost-sensitive learning, and modern deep learning approaches.

Finally, the framework studied here is specific to tabular data. Class imbalance in image classification is typically addressed through different mechanisms—gated convolutional architectures with hybrid connectivity [59], focal-loss reweighting, and end-to-end representation learning that subsumes the separate overlap-mitigation step into the network itself—and the empirical conclusions drawn here do not translate directly to those settings. A parallel study of imbalanced image classification that compares architectural choices against augmentation-based mitigation would be a useful follow-up, but lies outside the scope of the present work.

One further caveat applies to Section 5.3. The $F1 > 0.70$ cutoff fails as a predictive rule on held-out data, with validation accuracy at 39.6%, and we therefore report the underlying pattern (mean $\Delta F1$ more negative at high baseline F1) only at the aggregate descriptive level. Practitioners who attempt to use the cutoff as a per-dataset decision rule should not expect it to outperform chance.

5.5. Sensitivity to Class Weighting

One concern about the strong-classifier advantage demonstrated above is whether it reflects a hidden mitigation step embedded in `class_weight = 'balanced'` rather than a property of the classifiers themselves. Class-weighting is a within-classifier configuration setting, not a data-level preprocessing step, but it does redistribute the loss surface and therefore deserves separate treatment. Table 10 reports the baseline F1 with and without the class weighting argument across the 58 datasets. Removing it collapses the linear classifiers—SVM drops by 25.7 F1 points and Logistic Regression by 13.7—but leaves the four tree ensembles within ± 3.5 points of their weighted baseline (Random Forest

in fact gains 1.7 points). The Neural Network, XGBoost, k-NN, and Naive Bayes are unaffected because the default protocol for those classifiers does not use the weighting argument. The strong classifiers therefore remain at the top of the ranking regardless of whether class weights are applied, which means the dual-baseline advantage reported in Sections 4.3 and 4.4 is a property of the classifier choice rather than of a hidden weighting step inside the baseline configuration.

Table 10. Baseline F1 with and without `class_weight = 'balanced'` (mean across 58 datasets). Δ is the change introduced by removing class weights from classifiers that accept the parameter.

Classifier	Baseline F1 (Weighted)	Baseline F1 (No Class-Weight)	Δ
LightGBM	0.599	0.565	−0.034
Neural Network	0.592	0.592	0.000
Decision Tree	0.568	0.560	−0.008
Extra Trees	0.563	0.558	−0.005
XGBoost	0.560	0.560	0.000
SVM	0.554	0.297	−0.257
k-NN	0.552	0.552	0.000
Random Forest	0.546	0.562	+0.017
Logistic Regression	0.372	0.235	−0.137
Naive Bayes	0.206	0.206	0.000

6. Conclusions

The central finding pivots on the choice of comparison baseline. Measured against the same classifier without preprocessing, mitigation helps: a mean gain of +4.13 points, with oversampling methods succeeding in roughly 64% of configurations for weak classifiers. Measured against a strong-classifier baseline, the result reverses to a mean of −4.64 points, with 43.2% of configurations reaching or surpassing the honest LODO baseline. The two numbers measure different things and do not constitute an inconsistency. Mitigation narrows the gap between weak and strong classifiers, but it does not close it.

The variance decomposition quantifies the relative importance of each factor. Classifier identity accounted for $\eta^2 = 0.157$ of F1 variance; the mitigation technique accounted for $\eta^2 = 0.010$ —approximately a sixteen-fold gap among controllable factors. Across 10 classifiers, 11 techniques, 58 datasets, and 6960 configurations, the choice of algorithm consistently outweighed the choice of preprocessing. Dataset characteristics carried the largest share at $\eta^2 = 0.620$, placing the dominant driver of F1—what problem is being analyzed—outside both methodological levers.

Ceiling effects do not explain the result. The low stratum—datasets where the honest LODO baseline F1 was 0.48, leaving 51.6% of the metric range open—still showed mean Δ F1 of −2.03% under mitigation ($p < 0.001$). Degradation was larger on easier datasets than on harder ones. The perturbation cost of preprocessing is most apparent when the ensemble is otherwise performing well.

We reached the conclusion that in binary classification problems with moderate overlap: skip preprocessing, proceed with a strong tree-ensemble classifier (the honest LODO selector returned LightGBM on every dataset in this collection; the remaining tree-ensembles—Random Forest, Extra Trees, and XGBoost—form a tightly clustered alternative pool on mean F1 between 0.546 and 0.599), and allocate effort to hyperparameter optimization. When deployment constraints force a weaker algorithm, SMOTE remains the least risky preprocessing option—but the ceiling it reaches still falls short of what an unconstrained ensemble achieves on raw data. Open questions for subsequent work include multi-class overlap dynamics, extreme overlap regimes above $N1 = 0.5$, and metric sensitivity beyond F1. The most useful future direction is a dataset-level decision

framework—evaluating the complexity profile of the data to guide between preprocessing and classifier selection upfront—which we develop in a companion work currently under review.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app16146905/s1>.

Author Contributions: Conceptualization, S.K. and R.Ü.; methodology, S.K. and R.Ü.; software, S.K.; validation, S.K. and R.Ü.; formal analysis, S.K.; investigation, S.K.; resources, S.K. and R.Ü.; data curation, S.K. and R.Ü.; writing—original draft preparation, S.K. and R.Ü.; writing—review and editing, S.K. and R.Ü.; visualization, S.K.; supervision, R.Ü.; project administration, R.Ü. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The 58 binary classification datasets used in this study are publicly available from the KEEL repository (<https://sci2s.ugr.es/keel/datasets.php> accessed on 10 February 2026). All experimental scripts are provided as Supplementary Materials and available at https://github.com/samikaya00/overlap_benchmark (accessed on 10 June 2026).

Acknowledgments: The authors used Claude (Claude Opus 4.6) [60] (Anthropic) as an AI assistant during the preparation of this work. Specifically, Claude was used to assist with (1) developing and debugging the Python (v3.12.12) scripts for the experimental pipeline (Sections 3 and 4), (2) reviewing and refining the manuscript text for clarity and grammar, and (3) translation of the content written in the native language of the authors. All AI-assisted outputs were reviewed, verified, and edited by the authors, who take full responsibility for the content of this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADASYN	Adaptive Synthetic Sampling
ANOVA	Analysis of Variance
AUC	Area Under the Curve
BorderlineSMOTE	Borderline Synthetic Minority Over-sampling Technique
CART	Classification and Regression Trees
CBO	Cluster-Based Oversampling
ENN	Edited Nearest Neighbors
F1	F1-score (harmonic mean of precision and recall)
IR	Imbalance Ratio
KEEL	Knowledge Extraction based on Evolutionary Learning
k-NN	k-Nearest Neighbors
LightGBM	Light Gradient-Boosting Machine
LODO	Leave-One-Dataset-Out
LOF	Local Outlier Factor
MLP	Multilayer Perceptron
MST	Minimum Spanning Tree
N1	Ho–Basu overlap measure (fraction of borderline instances)
N3	Leave-one-out nearest-neighbor error rate (complexity measure)
NCR	Neighborhood Cleaning Rule
OLS	Ordinary Least Squares
PR	Precision–Recall
RBF	Radial Basis Function

RENN	Repeated Edited Nearest Neighbors
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Over-sampling Technique
SMOTE+ENN	SMOTE combined with Edited Nearest Neighbors
SMOTE+Tomek	SMOTE combined with Tomek Links
SVM	Support Vector Machine
T1	Hypersphere-coverage complexity measure (Ho-Basu)
XGBoost	Extreme Gradient Boosting

Appendix A

Table A1. Characteristics of the 58 KEEL binary classification datasets used in this study. *n* is the total instance count, Features is the dimensionality, IR is the imbalance ratio, Minority is the count of minority-class instances, N1 is the Ho-Basu overlap measure (proportion of borderline instances), and N3 is the leave-one-out nearest-neighbor error rate. N1 and N3 are computed on the canonical mitigation_results table; instance and feature counts come from the KEEL repository metadata. Ranges across the 58 datasets: *n* ∈ [108, 1829], IR ∈ [5.0, 41.4], N1 ∈ [0.00, 0.42].

Dataset	n	Features	IR	Minority	N1	N3
03subcl5-600-5-0-BI	600	2	5.0	100	0.237	0.052
03subcl5-600-5-30-BI	600	2	5.0	100	0.375	0.118
03subcl5-600-5-50-BI	600	2	5.0	100	0.415	0.165
03subcl5-600-5-60-BI	600	2	5.0	100	0.395	0.170
03subcl5-600-5-70-BI	600	2	5.0	100	0.420	0.177
03subcl5-800-7-0-BI	800	2	7.0	100	0.181	0.041
03subcl5-800-7-30-BI	800	2	7.0	100	0.310	0.095
03subcl5-800-7-50-BI	800	2	7.0	100	0.330	0.136
03subcl5-800-7-60-BI	800	2	7.0	100	0.338	0.131
03subcl5-800-7-70-BI	800	2	7.0	100	0.347	0.160
04clover5z-600-5-0-BI	600	2	5.0	100	0.233	0.048
04clover5z-600-5-30-BI	600	2	5.0	100	0.315	0.117
04clover5z-600-5-50-BI	600	2	5.0	100	0.357	0.143
04clover5z-600-5-60-BI	600	2	5.0	100	0.393	0.167
04clover5z-600-5-70-BI	600	2	5.0	100	0.397	0.182
04clover5z-800-7-0-BI	800	2	7.0	100	0.191	0.049
04clover5z-800-7-30-BI	800	2	7.0	100	0.259	0.087
04clover5z-800-7-50-BI	800	2	7.0	100	0.309	0.130
04clover5z-800-7-60-BI	800	2	7.0	100	0.328	0.134
04clover5z-800-7-70-BI	800	2	7.0	100	0.331	0.147
cleveland-0_vs_4	173	13	12.6	13	0.130	0.056
ecoli-0-1-3-7_vs_2-6	281	7	39.1	7	0.078	0.032
ecoli-0-1-4-6_vs_5	280	6	13.0	20	0.036	0.021
ecoli-0-1-4-7_vs_2-3-5-6	336	7	10.6	29	0.122	0.054
ecoli-0-1-4-7_vs_5-6	332	6	12.3	25	0.090	0.036
ecoli-0-1_vs_5	240	6	11.0	20	0.046	0.029
ecoli4	336	7	15.8	20	0.051	0.018
glass-0-1-4-6_vs_2	205	9	11.1	17	0.239	0.112
glass-0-1-6_vs_2	192	9	10.3	17	0.276	0.120
glass-0-1-6_vs_5	184	9	19.4	9	0.076	0.022
glass-0-6_vs_5	108	9	11.0	9	0.065	0.009
glass2	214	9	11.6	17	0.243	0.112
glass4	214	9	15.5	13	0.121	0.033
glass5	214	9	22.8	9	0.065	0.019
led7digit-0-2-4-5-6-7-8-9_vs_1	443	7	11.0	37	0.147	0.038
page-blocks-1-3_vs_4	472	10	15.9	28	0.047	0.002
paw02a-600-5-0-BI	600	2	5.0	100	0.137	0.023

Table A1. Cont.

Dataset	n	Features	IR	Minority	N1	N3
paw02a-600-5-30-BI	600	2	5.0	100	0.250	0.097
paw02a-600-5-50-BI	600	2	5.0	100	0.288	0.127
paw02a-600-5-60-BI	600	2	5.0	100	0.320	0.128
paw02a-600-5-70-BI	600	2	5.0	100	0.350	0.173
paw02a-800-7-0-BI	800	2	7.0	100	0.110	0.019
paw02a-800-7-30-BI	800	2	7.0	100	0.215	0.083
paw02a-800-7-50-BI	800	2	7.0	100	0.258	0.102
paw02a-800-7-60-BI	800	2	7.0	100	0.281	0.104
paw02a-800-7-70-BI	800	2	7.0	100	0.302	0.129
shuttle-c0-vs-c4	1829	9	13.9	123	0.000	0.001
shuttle-c2-vs-c4	129	9	20.5	6	0.016	0.008
vowel0	988	13	10.0	90	0.009	0.000
yeast-0-2-5-6_vs_3-7-8-9	1004	8	9.1	99	0.253	0.080
yeast-1-2-8-9_vs_7	947	8	30.6	30	0.128	0.043
yeast-1-4-5-8_vs_7	693	8	22.1	30	0.185	0.062
yeast-1_vs_7	459	7	14.3	30	0.216	0.078
yeast-2_vs_4	514	8	9.1	51	0.128	0.051
yeast-2_vs_8	482	8	23.1	20	0.081	0.037
yeast4	1484	8	28.1	51	0.105	0.040
yeast5	1484	8	32.7	44	0.057	0.019
yeast6	1484	8	41.4	35	0.069	0.030

References

1. Vuttipittayamongkol, P.; Elyan, E.; Petrovski, A. On the class overlap problem in imbalanced data classification. *Knowl.-Based Syst.* **2021**, *212*, 106631. [\[CrossRef\]](#)
2. García, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Springer: Cham, Switzerland, 2015.
3. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. *Learning from Imbalanced Data Sets*; Springer: Cham, Switzerland, 2018.
4. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284. [\[CrossRef\]](#)
5. Thabtah, F.; Hammoud, S.; Kamalov, F.; Gonsalves, A. Data imbalance in classification: Experimental evaluation. *Inf. Sci.* **2020**, *513*, 429–441. [\[CrossRef\]](#)
6. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [\[CrossRef\]](#)
7. Tomek, I. Two modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **1976**, *6*, 769–772. [\[CrossRef\]](#)
8. Laurikkala, J. Improving identification of difficult small classes by balancing class distribution. In *Proceedings of the 8th Conference on Artificial Intelligence in Medicine in Europe, Cascais, Portugal, 1–4 July 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 63–66.
9. Kumar, A.; Singh, D.; Yadav, R.S. Class overlap handling methods in imbalanced domain: A comprehensive survey. *Multimed. Tools Appl.* **2024**, *83*, 63243–63290. [\[CrossRef\]](#)
10. López, V.; Fernández, A.; García, S.; Palade, V.; Herrera, F. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* **2013**, *250*, 113–141. [\[CrossRef\]](#)
11. Blagus, R.; Lusa, L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinform.* **2013**, *14*, 106. [\[CrossRef\]](#)
12. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181.
13. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [\[CrossRef\]](#)
14. Douzas, G.; Bacao, F.; Last, F. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Inf. Sci.* **2018**, *465*, 1–20. [\[CrossRef\]](#)
15. Krawczyk, B.; Woźniak, M.; Schaefer, G. Cost-sensitive decision tree ensembles for effective imbalanced classification. *Appl. Soft Comput.* **2014**, *14*, 554–562. [\[CrossRef\]](#)

16. Kubat, M.; Matwin, S. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the Fourteenth International Conference on Machine Learning, Nashville, TN, USA, 8–12 July 1997*; Morgan Kaufmann: San Francisco, CA, USA, 1997; pp. 179–186.
17. Barandela, R.; Sánchez, J.S.; García, V.; Rangel, E. Strategies for learning in class imbalance problems. *Pattern Recognit.* **2003**, *36*, 849–851. [\[CrossRef\]](#)
18. Zhou, Z.H.; Liu, X.Y. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 63–77. [\[CrossRef\]](#)
19. Fernández, A.; García, S.; Herrera, F.; Chawla, N.V. SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *J. Artif. Intell. Res.* **2018**, *61*, 863–905. [\[CrossRef\]](#)
20. Das, S.; Mullick, S.S.; Zelinka, I. On supervised class-imbalanced learning: An updated perspective and some key challenges. *IEEE Trans. Artif. Intell.* **2022**, *3*, 973–993. [\[CrossRef\]](#)
21. Elkan, C. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence, Seattle, WA, USA, 4–10 August 2001*; Morgan Kaufmann: San Francisco, CA, USA, 2001; Volume 17, pp. 973–978.
22. Dietterich, T.G. Ensemble methods in machine learning. In *Proceedings of the Multiple Classifier Systems. MCS 2000, Cagliari, Italy, 21–23 June 2000*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2000; Volume 1857, pp. 1–15.
23. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [\[CrossRef\]](#)
24. García, V.; Sánchez, J.S.; Mollineda, R.A.; Alejo, R.; Sotoca, J.M. The class imbalance problem in pattern classification and learning. In *Proceedings of the II Congreso Espanol de Informática (CEDI 2007)*; CEDI: Zaragoza, Spain, 2007; pp. 283–291.
25. Denil, M.; Trappenberg, T. Overlap versus imbalance. In *Proceedings of the Canadian Conference on Artificial Intelligence, Ottawa, ON, Canada, 31 May–2 June 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 220–231.
26. Prati, R.C.; Batista, G.E.A.P.A.; Monard, M.C. Class imbalances versus class overlapping: An analysis of a learning system behavior. In *Proceedings of the Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, 26–30 April 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 312–321.
27. Carvalho, M.; Pinho, A.J.; Brás, S. How hard is my imbalanced classification task? An explainable supervised approach. *Expert Syst. Appl.* **2025**, *268*, 130695.
28. Santos, M.S.; Abreu, P.H.; Japkowicz, N.; Fernández, A.; Santos, J. A unifying view of class overlap and imbalance: Key concepts, multi-view panorama, and open avenues for research. *Inf. Fusion* **2023**, *89*, 228–253. [\[CrossRef\]](#)
29. Santos, M.S.; Abreu, P.H.; Japkowicz, N.; Fernández, A.; Santos, J. On the joint-effect of class imbalance and overlap: A critical review. *Artif. Intell. Rev.* **2022**, *55*, 6207–6275. [\[CrossRef\]](#)
30. Vuttipittayamongkol, P.; Papasratorn, B.; Elyan, E. Overlap and imbalance effect on class imbalance problem. In *Proceedings of the 9th International Conference on Advances in Information Technology, Hangzhou, China, 19–21 October 2018*; IEEE: New York, NY, USA, 2018; pp. 1–6.
31. Lorena, A.C.; Garcia, L.P.F.; Lehmann, J.; Souto, M.C.P.; Ho, T.K. How complex is your classification problem? A survey on measuring classification complexity. *ACM Comput. Surv.* **2019**, *52*, 1–34. [\[CrossRef\]](#)
32. Al Hosni, O.; Starkey, A. Investigating the performance of data complexity and instance hardness measures as a meta-feature in overlapping classes problem. In *Proceedings of the 2023 7th International Conference on Cloud and Big Data Computing, Manchester UK, 17–19 August 2023*; Association for Computing Machinery: New York, NY, USA, 2023.
33. Eberlein, J.; Rodriguez, D.; Harrison, R. The effect of data complexity on classifier performance. *Empir. Softw. Eng.* **2024**, *30*, 16. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern Classification*; Wiley: Hoboken, NJ, USA, 2012.
35. Ho, T.K.; Basu, M. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 289–300. [\[CrossRef\]](#)
36. Napierała, K.; Stefanowski, J. Types of minority class examples and their influence on learning classifiers from imbalanced data. *J. Intell. Inf. Syst.* **2016**, *46*, 563–597.
37. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 15–18 May 2000*; Association for Computing Machinery: New York, NY, USA, 2000; pp. 93–104.
38. Wilson, D.L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.* **1972**, *2*, 408–421. [\[CrossRef\]](#)
39. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proceedings of the International Conference on Intelligent Computing, Hefei, China, 23–26 August 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 878–887.

40. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks, Hong Kong, China, 1–8 June 2008*; IEEE: Piscataway, NJ, USA, 2008; pp. 1322–1328.
41. Jo, T.; Japkowicz, N. Class imbalances versus small disjuncts. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 40–49. [\[CrossRef\]](#)
42. Sun, P.; Du, Y.; Xiong, S. Nearest neighbors and density-based undersampling for imbalanced data classification with class overlap. *Neurocomputing* **2024**, *609*, 128492. [\[CrossRef\]](#)
43. Khan, A.A.; Chaudhari, O.; Chandra, R. A review of ensemble learning and data augmentation models for class imbalanced problems. *Expert Syst. Appl.* **2024**, *244*, 122778. [\[CrossRef\]](#)
44. Cerqueira, V.; Torgo, L.; Branco, P. Automated imbalanced classification via layered learning. *Mach. Learn.* **2023**, *112*, 2083–2104.
45. Rice, J.R. The algorithm selection problem. *Adv. Comput.* **1976**, *15*, 65–118. [\[CrossRef\]](#)
46. Rivolli, A.; Garcia, L.P.F.; Soares, C.; Vanschoren, J.; de Carvalho, A.C.P.L.F. Meta-features for meta-learning. *Knowl.-Based Syst.* **2022**, *240*, 108101. [\[CrossRef\]](#)
47. Galar, M.; Fernández, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.* **2012**, *42*, 463–484. [\[CrossRef\]](#)
48. Zhang, J.; Mani, I. k-NN approach to unbalanced data distributions: A case study involving information extraction. In *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Datasets, Washington, DC, USA, 21 July 2003*; Association for Computing Machinery: New York, NY, USA, 2003.
49. Dar, A.H.; Farooq, M.A. Handling class overlap using overlap driven under-sampling with balanced random forest. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 102059. [\[CrossRef\]](#)
50. D'souza, A.; Swetha, M.; Sarawagi, S. Synthetic tabular data generation for imbalanced classification: The surprising effectiveness of an overlap class. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025*; Association for Computing Machinery: New York, NY, USA, 2025; Volume 39, pp. 16127–16134.
51. Haluška, R.; Brabec, J. Benchmark of data preprocessing methods for imbalanced classification. In *Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2023*; IEEE: Piscataway, NJ, USA, 2023.
52. Moniz, N.; Cerqueira, V. Automated imbalanced classification via meta-learning. *Expert Syst. Appl.* **2021**, *178*, 115011. [\[CrossRef\]](#)
53. Alcalá-Fdez, J.; Fernández, A.; Luengo, J.; Derrac, J.; García, S.; Sánchez, L.; Herrera, F. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Mult.-Valued Log. Soft Comput.* **2011**, *17*, 255–287.
54. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **2017**, *18*, 1–5.
55. Verikas, A.; Gelzinis, A.; Bacauskiene, M. Mining data with random forests: A survey and results of new tests. *Pattern Recognit.* **2011**, *44*, 330–349. [\[CrossRef\]](#)
56. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
57. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
58. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794.
59. Yang, C.; An, Z.; Zhu, H.; Hu, X.; Zhang, K.; Xu, K.; Li, C.; Xu, Y. Gated convolutional networks with hybrid connectivity for image classification. In *Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020*; AAAI: Menlo Park, CA, USA, 2020; Volume 34, pp. 12581–12588.
60. Anthropic. Claude Opus 4.6. 2026. Available online: <https://www.claude.ai> (accessed on 15 February 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.