Local distribution-based adaptive oversampling for imbalanced regression

Anonymous authors
Paper under double-blind review

Abstract

Imbalanced regression occurs when continuous target variables have skewed distributions, creating sparse regions that are difficult for machine learning models to predict accurately. This issue particularly affects neural networks, which often struggle with imbalanced data. While class imbalance in classification has been extensively studied, imbalanced regression remains relatively unexplored, with few effective solutions. Existing approaches often rely on arbitrary thresholds to categorize samples as rare or frequent, ignoring the continuous nature of target distributions. These methods can produce synthetic samples that fail to improve model performance and may discard valuable information through undersampling. To address these limitations, we propose LDAO (Local Distribution-based Adaptive Oversampling), a novel data-level approach that avoids categorizing individual samples as rare or frequent. Instead, LDAO learns the global distribution structure by decomposing the dataset into a mixture of local distributions, each preserving its statistical characteristics. LDAO then models and samples from each local distribution independently before merging them into a balanced training set. LDAO achieves a balanced representation across the entire target range while preserving the inherent statistical structure within each local distribution. In extensive evaluations on 45 imbalanced datasets, LDAO outperforms state-of-the-art oversampling methods on both frequent and rare target values, demonstrating its effectiveness for addressing the challenge of imbalanced regression.

In classification tasks, imbalance occurs when some classes have far fewer samples than others, in both binary and multi-class settings (He & Garcia, 2009; Guo et al., 2017). The minority classes are overshadowed by the majority classes, causing traditional classifiers to focus on the more abundant classes and perform poorly in identifying minority classes (Chawla et al., 2002). This leads to inaccurate detection of minority classes since models struggle to recognize patterns in sparsely sampled regions (Buda et al., 2018). Critical applications like fraud detection, medical diagnostics, and fault detection are particularly affected, as rare but important classes might be missed (Johnson & Khoshgoftaar, 2019). Therefore, effective handling of imbalanced data is crucial for reliable performance across all classes, ensuring significant rare examples are properly detected (Liu et al., 2009).

While imbalance is widely studied in classification (He & Garcia, 2009), it also affects regression tasks that predict continuous values (Krawczyk, 2016). Imbalanced regression occurs when certain target ranges (typically rare or extreme cases) have significantly fewer samples than others (Branco et al., 2016; Torgo et al., 2013). Traditional regression models struggle to accurately predict these rare values because they focus on more frequent, well-represented ranges (Chawla et al., 2004; Branco et al., 2016). Accurately predicting these less frequent target ranges is challenging yet crucial for many real-world applications (Torgo et al., 2013).

Figure 1 contrasts imbalanced classification with imbalanced regression. In classification, identifying minority classes is straightforward because labels are discrete and can be counted. In regression, minority regions exist within continuous target distributions, often in sparsely populated tails. While classification imbalance relates to distinct labels, regression imbalance depends on distribution shape (skewed or multi-modal). Classification methods cannot be directly applied to regression since there is no simple way to count labels in

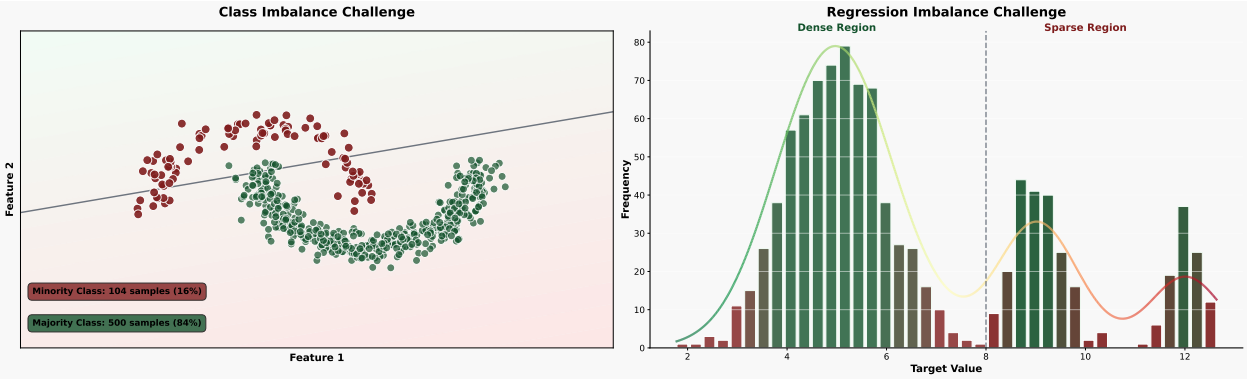


Figure 1: The left image shows a classification problem with easily identifiable minority (red) and majority (green) classes. The right image illustrates an imbalanced regression problem where target values in the sparse region (red regions) are underrepresented, making them more difficult to detect and accurately predict.

order to identify underrepresented values. This makes detecting and addressing sparse regions a significant challenge in imbalanced regression.

Imbalanced regression impacts many critical real-world applications. In genomic prediction, pathogenicity scores typically favour certain ranges, placing potentially harmful genetic variants in sparse distribution tails. Standard models struggle to predict these rare but important extremes (Kaur & Sarmadi, 2024).

For extreme wind event forecasting, measurements primarily cluster around moderate speeds, while infrequent high-speed events critical for hazard warnings and energy grid stability are often underestimated by conventional models that prioritize majority data (Scheepens et al., 2023). In healthcare, hospital length-of-stay data is heavily skewed toward short stays, with fewer patients requiring extended care. Models trained predominantly on brief stays underestimate resource needs for longer admissions, affecting capacity planning and staffing, particularly during crises like pandemics (Xu et al., 2022). Similar challenges exist in economics, finance, and engineering, where accurately predicting rare but high-impact events is essential for effective decision-making.

Several solutions have been proposed to address imbalanced regression at both data and algorithmic levels. Some extend SMOTE from classification to regression by generating synthetic examples in the target distribution's tails, increasing representation of rare values (Torgo et al., 2013; Branco et al., 2017). Others apply cost-sensitive methods that penalize errors on underrepresented values more heavily, encouraging models to focus on these instances (Steininger et al., 2021). Despite showing promise, the current solutions for imbalanced regression remain limited, and more robust approaches are needed.

We propose LDAO, a novel data-level oversampling approach that addresses imbalanced regression through adaptive region-specific sampling. LDAO preserves the continuous nature of regression targets while effectively enhancing representation across the entire target distribution. Our approach provides a more robust foundation for regression models when dealing with imbalanced target distributions.

1 Related Work

Early approaches to imbalanced data focused on data-level methods, particularly in classification, and data-level strategies appear in nearly one-third of imbalanced classification papers, though most were designed for discrete classes (Guo et al., 2017). These methods rebalance datasets by either adding minority samples or removing majority samples (Chawla et al., 2002; He et al., 2008). SMOTE exemplifies this approach by generating synthetic minority samples through interpolation in feature space (Chawla et al., 2002). It works by selecting a minority sample, finding its k-nearest minority neighbors, and creating new points by

randomly interpolating between their feature values. This technique helps distribute minority samples more evenly throughout the data space.

Several SMOTE variations have emerged to address specific challenges. Borderline-SMOTE focuses on minority samples near class boundaries (Han et al., 2005), while DBSMOTE leverages local density information to generate synthetic points in regions where minority classes are concentrated (Bunkhumpornpat et al., 2012). These methods aim to improve minority class coverage and enhance classifier performance on imbalanced datasets.

Adapting data-level strategies to regression is more difficult because the target variable is continuous (He & Ma, 2013; Krawczyk, 2016). Early approaches to imbalanced regression modified classification resampling by introducing a relevance function $\phi(y)$ to distinguish between "rare" and "frequent" target regions (Torgo & Ribeiro, 2007). This function assigns higher scores to more extreme target values, while a user-specified threshold determines which points are rare versus common (Torgo & Ribeiro, 2007; Ribeiro, 2011). Using this relevance function framework, Torgo et al. developed two main resampling methods for regression Torgo et al. (2013; 2015).

The first method applies random undersampling to remove samples with low $\phi(y)$ scores (common target values), reducing their overrepresentation. The second method, SMOTER, generates new samples in regions with high $\phi(y)$ scores (rare target values) by interpolating feature values of nearby rare samples and calculating target values as weighted averages of these neighbors (Torgo et al., 2013; 2015). While these approaches were pioneering in addressing regression imbalance and demonstrated that focusing on rare targets improves performance (Torgo et al., 2015), they depend on user-chosen rarity thresholds that may be arbitrary and fail to reflect the true structure of continuous target distributions (Branco et al., 2019).

SMOGN (Synthetic Minority Over-sampling Technique for Regression with Gaussian Noise) extends SMOTER by combining interpolation with noise-based oversampling. For each rare instance, SMOGN generates either an interpolated synthetic point (as in SMOTER) or adds Gaussian noise to create a perturbed point (Branco et al., 2017). This hybrid approach provides more flexible oversampling of sparse regions and outperforms SMOTER (Branco et al., 2017). SMOGN also applies undersampling to reduce overrepresented frequent examples, establishing it as a state-of-the-art resampling method for imbalanced regression (Branco et al., 2017).

Additional oversampling approaches include simple random oversampling (replicating high-ϕ examples) and pure noise-based oversampling (adding small random perturbations to rare examples) (Branco et al., 2019). Another method, WERCS (Weighted Relevance-based Combination Strategy), probabilistically selects instances for oversampling or undersampling based on relevance scores: rare instances have higher chances of duplication while common instances are typically undersampled unless kept for diversity. This unified approach blends techniques to avoid hard threshold cutoffs (Branco et al., 2019).

More recently, Camacho et al. developed Geometric SMOTE (G-SMOTE) for regression, adapting geometrically enhanced SMOTE to continuous targets Camacho et al. (2022). G-SMOTE generates synthetic samples in minority regions by considering both feature-space interpolation and target distribution geometry, offering greater flexibility in sample placement (Camacho et al., 2022).

Stocksieker et al. combined weighted resampling and data augmentation to better represent covariate distributions and reduce overfitting Stocksieker et al. (2023). Their approach weights underrepresented regions while using noise addition or interpolation to cover wider data ranges. Camacho and Bacao introduced WSMOTER (Weighting SMOTE for Regression), replacing fixed thresholds with instance-based weighting to better highlight underrepresented targets and improve synthetic sample generation in sparse regions Camacho & Bacao (2024). Stocksieker et al. later developed GOLIATH, a framework handling both noise and interpolation methods with hybrid generators and wild-bootstrap steps for continuous targets Stocksieker et al. (2024). Aleksic and García-Remesal proposed SUS (Selective Under-Sampling), which selectively removes only those majority samples that provide little additional information, along with an iterative variant (SUSiter) that reintroduces discarded samples over time, showing strong results on high-dimensional datasets Aleksic & García-Remesal (2025).

In addition to data-level strategies, algorithm-level solutions embed imbalance handling directly into the learning process. Cost-sensitive learning exemplifies this approach by modifying loss functions or instance weights to make models focus on rare targets and incur higher penalties for errors on these examples. This concept originated in classification by weighting classes inversely to their frequency and extends naturally to regression by emphasizing rare or extreme target values (Zhou & Liu, 2010; Elkan, 2001; Domingos, 1999). By imposing higher costs on underrepresented outcomes, these approaches drive models to reduce errors where accuracy matters most.

DenseLoss is a cost-sensitive approach designed specifically for imbalanced regression (Steininger et al., 2021). It uses DenseWeight, a density-based weighting strategy that assigns higher weights to samples with less frequent target values. This directs the model to focus on rare or extreme cases where accuracy is most critical (Steininger et al., 2021). DenseWeight estimates the target distribution's probability density and gives larger weights to low-density targets, scaling the loss function during training to emphasize prediction accuracy on rare cases. Unlike oversampling methods, DenseLoss modifies the learning process without creating or removing examples from the training set.

Yang et al. proposed a framework for deep imbalanced regression that improves prediction performance for extreme target values using label-distribution smoothing and adaptive sampling Yang et al. (2021). Their work demonstrates how continuous target imbalance can severely impact deep neural network performance. Ren et al. introduced Balanced MSE, which modifies standard mean-squared error to give higher weight to errors on rare targets Ren et al. (2022). Validated across multiple computer vision tasks, their approach shows that directly addressing label imbalance outperforms standard MSE in skewed settings.

While cost-sensitive approaches like DenseLoss show promise by avoiding synthetic data augmentation (Steininger et al., 2021), they remain relatively uncommon in imbalanced regression. A key challenge is their sensitivity to the weighting function—optimization can become unstable if rare cases are weighted too heavily (He & Garcia, 2009).

Ensemble learning has been explored for imbalanced regression challenges. While boosting and bagging improve model generalization by combining multiple learners, they don't directly address skewed distributions (Hoens & Chawla, 2013). Therefore, researchers typically combine ensemble methods with specific imbalance-countering techniques. One effective approach applies resampling within each ensemble iteration. Resampled Bagging, for instance, uses balanced subsets or weighted samples in each bootstrap replicate, preserving diversity across learners while reducing bias toward majority targets (Branco et al., 2019).

Moniz et al. extended boosting for rare targets through SMOTEBoost for regression, which generates synthetic samples in extreme target ranges and then uses boosting to emphasize these difficult-to-predict points Moniz et al. (2018). This combination of oversampling and ensemble methods proves especially valuable when models need to accurately capture tail values or outliers that standard regression approaches frequently miss.

Evaluation metrics for imbalanced regression have evolved alongside data-level, algorithmic-level, and ensemble-based solutions. Traditional metrics like MSE or MAE are often dominated by errors on frequent cases, prompting the development of specialized metrics that better assess performance on rare targets. The SERA metric (Squared Error Relevance Area) exemplifies this approach by incorporating relevance into evaluation, weighting errors according to the rareness of target values (Ribeiro & Moniz, 2020). Ribeiro and Moniz argue that SERA offers more informative model comparisons in imbalanced domains by highlighting performance on extreme values relative to common ones (Ribeiro & Moniz, 2020). While SERA itself is an evaluation metric rather than a training method, its development reflects the field's increasing focus on accurately assessing how well models handle rare events.

2 Motivation

Despite recent advances, imbalanced regression still faces significant challenges. Current methods often identify samples as rare or frequent and oversample the distribution based on that assumption. In many cases, these assumptions might be oversimplifications due to the complexity of continuous distributions and might not help the model generalize better or improve its prediction on rare samples. Also, many approaches try to oversample rare targets and undersample frequent targets, which distorts the nature of the dataset.

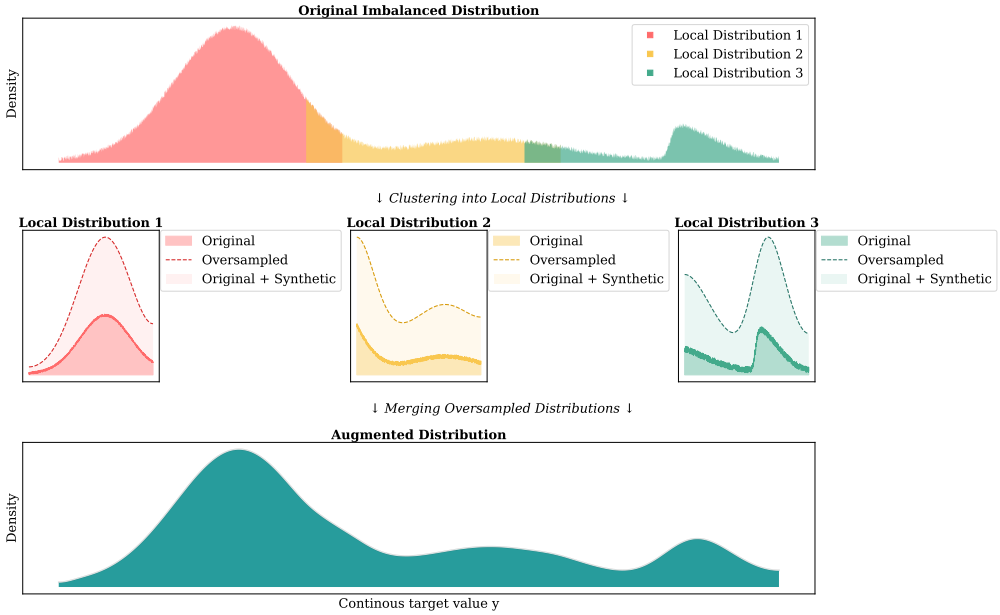


Figure 2: LDAO process overview. First, the imbalanced dataset (top) is decomposed into clusters. Then, each cluster is oversampled individually using kernel density estimation (middle). Lastly, these balanced clusters are combined into one dataset (bottom).

Removing information from datasets and labeling certain ranges as rare or frequent is problematic, as rare samples can occur in any range within the global distribution, especially in multi-modal distributions. Often, distributions have latent characteristics which make it difficult to simply classify which samples are rare or frequent.

In addition, interpolation and noise-based oversampling techniques may generate synthetic samples that don't reflect actual data patterns, especially in extremely sparse regions (Branco et al., 2019). Creating synthetic feature-target pairs that remain consistent with the original distribution is challenging, and most methods focus narrowly on extreme tails rather than the entire distribution. Many approaches rely heavily on relevance functions and thresholds that require careful parameter selection. Poor choices can lead to oversampling marginally rare areas while missing truly rare regions. This dependence on domain-specific tuning limits broader applicability across different datasets. Real-world applications in healthcare, genomics, environmental science, finance, and engineering frequently involve imbalanced regression tasks, and we need more robust approaches that tackle the problem of imbalanced regression as we have seen in classification.

Adaptive oversampling techniques in classification, such as the method by Wang et al. (2020), which leverages local distribution information to generate synthetic minority instances, demonstrate the potential of these strategies. However, these methods rely on discrete class labels and fixed thresholds that do not translate well to regression tasks, where the target variable is continuous.

Our method, LDAO, addresses these limitations by preserving data structure across all target ranges. Instead of using global thresholds or discarding frequent samples, LDAO augments each cluster of the target distribution independently, generating synthetic data that match local distribution patterns. Our goal in LDAO is to improve the model's prediction of rare samples while enhancing its overall generalization. LDAO aims to mitigate the trade-off between emphasizing rare samples and preserving overall generalization ability.

3 LDAO Method

We now describe LDAO, our local distribution-based adaptive oversampling method that effectively addresses the challenges of imbalanced regression. LDAO decomposes the global distribution using k-means clustering.

This clustering is applied simultaneously to both input features and target values, creating a mixture of local distributions in a joint feature–target space.

After clustering, we apply Gaussian kernel density estimation (KDE) within each cluster to model the local distribution independently and generate synthetic data points that closely match the cluster's actual distribution. By drawing samples independently within each cluster, LDAO avoids creating unrealistic synthetic points. Finally, we merge these augmented clusters into a single augmented dataset. The entire workflow is summarized in Figure 2: data clustering, local density estimation for synthetic data generation, and merging to achieve an augmented overall distribution.

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents features and $y_i \in \mathbb{R}$ represents the target variable, imbalance typically manifests as regions with fewer observations, making these areas difficult to model accurately. LDAO models the joint distribution of features and targets. Each data point $(\mathbf{x}_i, y_i)$ is embedded in the combined feature-target space as $\mathbf{z}_i = (\mathbf{x}_i, y_i) \in \mathbb{R}^{d+1}$. The overall joint distribution $P_{X,Y}$ is approximated by a mixture model composed of localized distributions:

$$P_{X,Y}(\mathbf{x}, y) \approx \sum_{k=1}^{K} \pi_k \, P_k(\mathbf{x}, y)$$

where π_k denotes the proportion of data in cluster k, and P_k characterizes the local distribution of that cluster.

3.1 Clustering in the Joint Feature-Target Space

We apply the standard k-means clustering algorithm to the dataset to identify natural clusters in the joint feature–target space. Figure 3 illustrates clustering for a sample data set, projected onto the first three principal components, and shows how points with similar features and target values are grouped for more precise local density modelling. For illustration purposes only, we chose 3 as the number of clusters. In this method, every data point $\mathbf{z}_i$ is assigned to one of K clusters. The goal is to find the best set of cluster centroids $\{\boldsymbol{\mu}_k\}_{k=1}^K$ and assignments $\{c_i\}_{i=1}^N$ such that the total squared Euclidean distance between the data points and their assigned centroids is minimized. Mathematically, we solve:

$$\underset{\{\boldsymbol{\mu}_k\}_{k=1}^K, \{c_i\}_{i=1}^N}{\arg\min} \sum_{i=1}^{N} \|\mathbf{z}_i - \boldsymbol{\mu}_{c_i}\|^2.$$

This formulation ensures that points with similar characteristics are grouped together (Sugar & James, 2003; Ikotun et al., 2023).

3.2 Determining the Number of Clusters

The performance of k-means clustering largely depends on the chosen number of clusters K. To choose an optimal K, we evaluate the Sum of Squared Errors (SSE), defined as:

$$\mathrm{SSE}(K) = \sum_{k=1}^{K} \sum_{\mathbf{z} \in \mathcal{D}_k} \|\mathbf{z} - \boldsymbol{\mu}_k\|^2,$$

where $\mathcal{D}_k$ represents the set of data points in cluster k. As we increase K, the SSE typically decreases because each point is closer to its cluster center.

To mitigate the pitfalls of the trade-off between oversimplification and excessive clustering, we use the elbow method (Syakur et al., 2018; Nainggolan et al., 2019). This method involves calculating the relative reduction in SSE when increasing the number of clusters. We define the relative improvement $\Delta(K)$ as:

$$\Delta(K) = \frac{\mathrm{SSE}(K-1) - \mathrm{SSE}(K)}{\mathrm{SSE}(K-1)}, \quad K = 2, \ldots, K_{\max}.$$

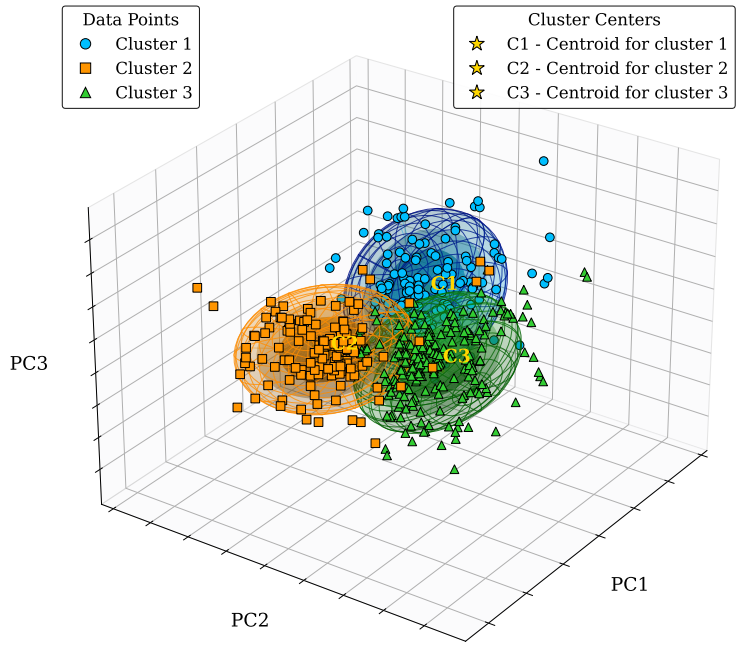


Figure 3: K-means clustering in the joint feature–target space for the Boston dataset (Harrison & Rubinfeld, 1978), projected onto the first three principal components. For illustration only, points are projected onto the first three principal components; clustering itself is done in the full $(d+1)$-dimensional space. Data points are grouped into three clusters (each indicated by a unique marker and color). Ellipsoidal density contours characterize the spread and orientation of each cluster. The centroids, marked with prominent gold symbols, represent the mean positions of the data points within their respective clusters.

The optimal number of clusters (K^*) is typically found where there is a significant drop in $\Delta(K)$. One important note is that at the elbow point, adding more clusters does not considerably reduce the Sum of Squared Errors (SSE) and might cause overfitting by capturing noise rather than true underlying patterns (Syakur et al., 2018; Nainggolan et al., 2019).

3.3 Kernel Density Estimation in Each Cluster

After the clustering step, we model the local distributions within each cluster separately using kernel density estimation (KDE) (Silverman, 1986), as illustrated in Figure 4.

This approach preserves each cluster's local structure, maintaining accurate relationships between features and targets without mixing patterns across clusters. Let each cluster $\mathcal{D}_k$ contain n_k data points in the joint feature-target space:

$$\mathcal{D}_k = \{\mathbf{z}_1^{(k)}, \mathbf{z}_2^{(k)}, \ldots, \mathbf{z}_{n_k}^{(k)}\},$$

where each data point is defined as $\mathbf{z}_j^{(k)} = (\mathbf{x}_j^{(k)}, y_j^{(k)}) \in \mathbb{R}^{d+1}$. We estimate the local density of cluster k using KDE with a Gaussian kernel as follows:

$$\hat{f}_k(\mathbf{z}) = \frac{1}{n_k} \sum_{j=1}^{n_k} K(\mathbf{z} - \mathbf{z}_j^{(k)}), \tag{1}$$

where the Gaussian kernel $K(\cdot)$ is defined by:

$$K(\mathbf{u}) = \frac{1}{(2\pi)^{\frac{d+1}{2}} |\mathbf{H}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{u}^\top \mathbf{H}^{-1}\mathbf{u}\right), \quad \mathbf{u} \in \mathbb{R}^{d+1}.$$

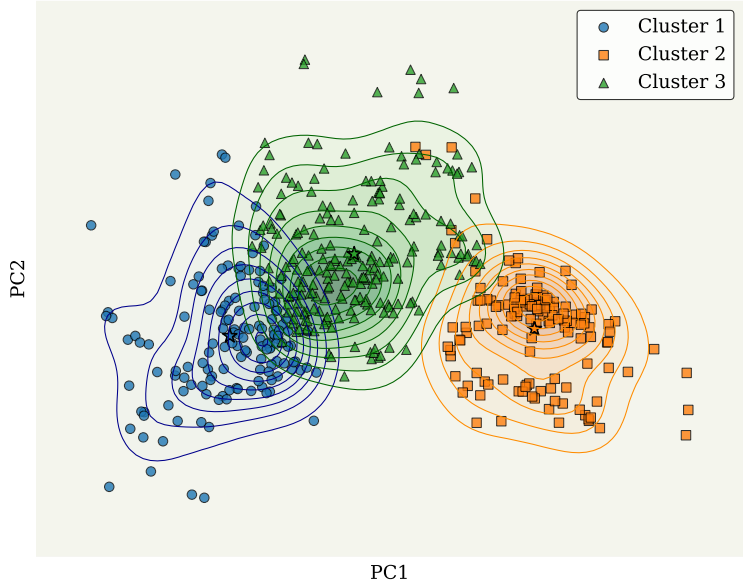


Figure 4: Clusters obtained via k-means on the feature–target data are projected onto two principal components. For each cluster, kernel density estimation (KDE) is performed and the overlaid contour lines represent the density gradients of the data in the reduced space.

Here, $\mathbf{H}$ is a $(d+1) \times (d+1)$ bandwidth (covariance) matrix that controls the shape and smoothness of the density estimate (Scott, 2015). The bandwidth matrix $\mathbf{H}$ determines how closely the estimated density follows the observed data points. A smaller determinant $|\mathbf{H}|$ results in sharply peaked density estimates localized around data points, while a larger determinant yields smoother densities.

Selecting an appropriate bandwidth matrix $\mathbf{H}$ is critical to KDE performance (Sheather & Jones, 1991). If $|\mathbf{H}|$ is too small, KDE tends to overfit noise, resulting in unrealistic density estimates. Conversely, an excessively large $|\mathbf{H}|$ oversmooths important local features. Practical approaches such as cross-validation, plug-in estimators, or generalized Silverman's rule for multivariate KDE can be employed independently for each cluster (Zhang et al., 2006). By fitting KDE independently to each cluster using carefully selected bandwidth matrices, we create localized models that accurately represent each region's unique distribution. This ensures that synthetic data generated in the subsequent steps reflect genuine local patterns, effectively addressing imbalance without distorting critical data relationships.

3.4 Oversampling Clusters

Once the density of each cluster has been estimated, we independently oversample each cluster. Figure 5 illustrates this process. Each cluster k, originally containing n_k points, is expanded by generating synthetic data points until reaching a target size defined as:

$$n'_k = \lceil \alpha_k n_k \rceil,$$

where the multiplier $\alpha_k > 1$ is a parameter of the algorithm that determines how much each cluster grows. We generate exactly $n'_k - n_k$ synthetic points $\mathbf{z}^*$ drawn from the local KDE-based density estimate $\hat{f}_k(\mathbf{z})$. The multiplier α_k can be uniform (identical across clusters) or adaptive (higher for clusters that are smaller or less dense). Typically, smaller or sparser clusters receive a larger multiplier, ensuring balanced representation throughout the dataset.

Synthetic points $\mathbf{z}^*$ are generated directly from the KDE-based local distribution defined in Equation equation 1:

$$\mathbf{z}^* \sim \hat{f}_k(\mathbf{z}).$$

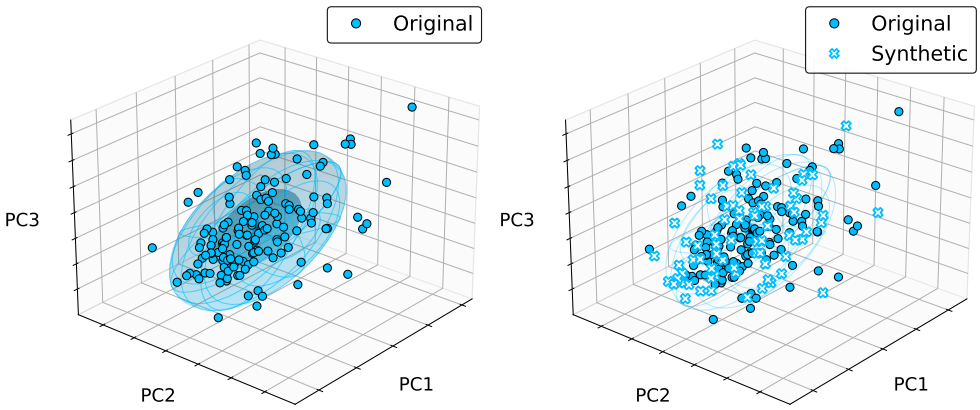


Figure 5: Each cluster is independently oversampled based on its own density estimate, ensuring that both sparse and dense areas are adequately represented without altering their local characteristics; the same procedure is applied to the remaining clusters, with cluster 1 displayed here.

For KDE with a Gaussian kernel using bandwidth matrix $\mathbf{H}$, synthetic samples are generated by randomly selecting an existing cluster data point and adding a Gaussian perturbation scaled by the Cholesky decomposition of the bandwidth matrix:

$$\mathbf{z}^* = \mathbf{z}_j^{(k)} + \mathbf{H}^{1/2}\boldsymbol{\epsilon}, \quad \text{where} \quad j \sim \text{Uniform}\{1, \ldots, n_k\}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{d+1}).$$

Here, $\mathbf{H}^{1/2}$ is the matrix square root (typically via Cholesky decomposition) of the bandwidth covariance matrix $\mathbf{H}$. This procedure accurately and efficiently samples synthetic points from the KDE estimate, reflecting the local structure and covariance of each cluster(Silverman, 1986; Scott, 2015).

3.5 Merging Augmented Clusters

After independently oversampling each cluster, the resulting augmented clusters are merged into a single balanced dataset, as illustrated in Figure 6. Formally, each augmented cluster is defined as:

$$\hat{\mathcal{D}}_k = \mathcal{D}_k \cup \mathcal{D}_k^*,$$

where $\mathcal{D}_k^*$ represents the synthetic samples generated for cluster k. The final augmented dataset $\hat{\mathcal{D}}$ is constructed by merging all augmented clusters:

$$\hat{\mathcal{D}} = \bigcup_{k=1}^{K} \hat{\mathcal{D}}_k.$$

The size of each cluster after augmentation is explicitly given by:

$$|\hat{\mathcal{D}}_k| = n'_k = \lceil \alpha_k n_k \rceil.$$

Hence, the total size of the merged dataset becomes:

$$|\hat{\mathcal{D}}| = \sum_{k=1}^{K} n'_k = \sum_{k=1}^{K} \left(n_k + (n'_k - n_k)\right).$$

By carefully selecting the multipliers α_k, we precisely control the number of synthetic points contributed by each cluster. This yields an augmented representation across the entire range of frequent and rare target values. The complete LDAO procedure is presented in Algorithm 1, which outlines all four steps of our proposed method.

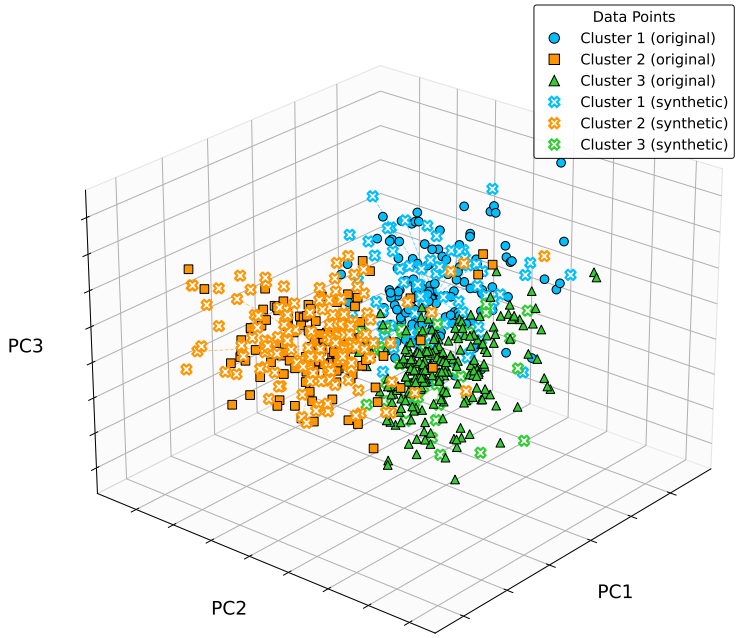


Figure 6: Visualization of original (filled markers) and synthetic (outlined markers) points in each cluster after local oversampling. The merged dataset is augmented while preserving all original data.

Algorithm 1 Local distribution–based adaptive oversampling (LDAO)

Input: data $\{(x_i, y_i)\}_{i=1}^{N}$, $K_{\min}, K_{\max}$, factors $\{\alpha_k\}$
Output: augmented set $\widehat{\mathcal{D}}$
Embed $z_i \leftarrow (x_i,\, y_i)$
for $K = K_{\min}$ **to** $K_{\max}$ **do**
    Cluster $\{z_i\} \rightarrow \{\mathcal{D}_k\}$; compute SSE($K$)
end for
Select K^* via elbow; recluster into $\{\mathcal{D}_k\}_{k=1}^{K^*}$
for $k = 1$ **to** K^* **do**
    Fit KDE f_k on $\mathcal{D}_k$
    Sample $\lceil \alpha_k\, |\mathcal{D}_k| \rceil - |\mathcal{D}_k|$ points from f_k to form $\widehat{\mathcal{D}}_k$
end for
$\widehat{\mathcal{D}} \leftarrow \bigcup_{k=1}^{K^*} \widehat{\mathcal{D}}_k$
return $\widehat{\mathcal{D}}$

4 Evaluation methodology

We evaluated LDAO with current state-of-the-art imbalanced regression approaches using a diverse collection of benchmark datasets. This section details our experimental design, including dataset selection, baseline methods, implementation specifics, hyperparameter tuning, and validation procedures.

Table 1: Dataset characteristics showing number of instances, features, and size category.

Dataset	Instances	Features	Size	Dataset	Instances	Features	Size
A1	198	11	Small	DEBUTANIZER	2394	7	Large
A2	198	11	Small	DEE	365	6	Small
A3	198	11	Small	DIABETES	43	2	Small
A7	198	11	Small	ELE-1	495	2	Small
ABALONE	4177	8	Large	ELE-2	1056	4	Medium
ACCELERATION	1732	14	Medium	FORESTFIRES	517	12	Medium
AIRFOILD	1503	5	Medium	FRIEDMAN	1200	5	Medium
ANALCAT	450	11	Small	FUEL	1764	37	Medium
AUTOMPG6	392	5	Small	HEAT	7400	11	Large
AUTOMPG8	392	7	Small	HOUSE	22784	16	Large
AVAILABLE_POWER	1802	15	Medium	KDD	316	18	Small
BASEBALL	337	16	Small	LASER	993	4	Medium
BOSTON	506	13	Medium	LUNGCANCER	442	24	Small
CALIFORNIA	20640	8	Large	MACHINECPU	209	6	Small
COCOMO	60	56	Small	CONCRETE_STRENGTH	1030	8	Medium
COMPACTIV	8192	21	Large	META	528	65	Medium
MORTGAGE	1049	15	Medium	MAXIMAL_TORQUE	1802	32	Medium
PLASTIC	1650	2	Medium	CPU	8192	12	Large
POLE	14998	26	Large	TRIAZINES	186	60	Small
QUAKE	2178	3	Large	WANKARA	1609	9	Medium
SENSORY	576	11	Medium	WINE_QUALITY	1143	12	Medium
STOCK	950	9	Medium	WIZMIR	1461	9	Medium
TREASURY	1049	15	Medium				

4.1 Datasets

We evaluated our method using 45 datasets from three sources: the Keel repository (Alcalá-Fdez et al., 2011), the collection at `https://paobranco.github.io/DataSets-IR` (Branco et al., 2019), and the repository at `https://github.com/JusciAvelino/imbalancedRegression` (Avelino et al., 2024).

These datasets span multiple domains and vary in size, dimensionality, and degree of imbalance, making them standard benchmarks for the rigorous evaluation of imbalanced regression methods and enabling fair comparisons with existing approaches. Table 1 shows the number of instances and features for each dataset. For further analysis in the results section, we categorized the datasets based on their number of instances. Specifically, datasets with fewer than 500 instances are labeled as "Small", those with between 500 and 1999 instances as "Medium", and those with more than 1999 instances as "Large". This categorization facilitates a detailed evaluation of our method's performance relative to dataset size.

4.2 Metrics

We evaluate LDAO against state-of-the-art approaches using multiple metrics that measure performance on both frequent and rare target values.

4.2.1 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) measures the overall prediction accuracy by calculating the square root of the average squared difference between predicted values and actual observations:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{2}$$

where y_i represents the true target value and $\hat{y}_i$ the predicted value for the i-th instance. RMSE provides a general assessment of model performance but can be dominated by errors in densely populated regions.

4.2.2 Squared Error-Relevance Area (SERA)

This metric provides a flexible way to evaluate models under non-uniform domain preferences. Let $\phi(\cdot)\colon \mathcal{Y} \to [0,1]$ be a relevance function that assigns higher scores to more important (for example, rare or extreme) target values. Then for any relevance threshold t, let

$$D^t = \{(x_i, y_i) \mid \phi(y_i) \geq t\},$$

and define

$$SER_t = \sum_{(x_i,y_i)\in D^t} \left(\hat{y}_i - y_i\right)^2. \tag{3}$$

SERA then integrates this quantity over all $t \in [0, 1]$:

$$SERA = \int_0^1 SER_t \, dt = \int_0^1 \sum_{(x_i,y_i)\in D^t} \left(\hat{y}_i - y_i\right)^2 dt. \tag{4}$$

SERA weights prediction errors by $\phi(y_i)$, emphasizing performance on extreme values while still considering accuracy across the entire domain. This makes it well-suited for imbalanced regression, where predicting rare values accurately is crucial (Ribeiro & Moniz, 2020).

4.2.3 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is an alternative error metric that evaluates the overall prediction performance by averaging the absolute differences between the predicted and actual values:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left|y_i - \hat{y}_i\right|. \tag{5}$$

MAE is less sensitive to outliers compared to RMSE and provides a straightforward interpretation of the average prediction error.

4.3 Machine Learning Algorithms

Following the approach of (Steininger et al., 2021), we evaluated all methods using a Multi-Layer Perceptron (MLP) with three hidden layers (10 neurons each) and ReLU activations. The output layer uses linear activation for regression. We trained models for 1000 epochs using Adam optimizer with early stopping to prevent overfitting. We compared LDAO against four approaches: Baseline (no resampling, using the original imbalanced data), SMOGN (an extension of SMOTER that incorporates Gaussian noise during oversampling), G-SMOTE (Geometric SMOTE adapted for regression tasks, using geometric interpolation), and DenseLoss (a cost-sensitive approach that weights errors by target density). These methods represent the current state-of-the-art method for handling imbalanced regression.

Table 2: Hyperparameter search spaces for each compared method

Method	Hyperparameter	Range	Description
LDAO	K (candidate clusters)	[2, 6]	Number of clusters tested (optimal K is selected using the elbow method)
	Multiplier per cluster	[1.0, 3.0]	Oversampling factor
	Bandwidth per cluster	[0.1, 2.0]	KDE smoothing parameter
SMOGN	k	{3, 5, 7, 9}	Number of neighbors
	sampling_method	{extreme, balance}	Sampling approach
	rel_thres	[0.0, 1.0]	Relevance threshold
G-SMOTE	q_rare	[0.05, 0.25]	Quantile for rarity
	truncation_factor	[0.0, 0.9]	Geometric truncation
	deformation_factor	[0.0, 0.9]	Geometric deformation
	k_neighbors	[2, 5]	Number of neighbors
	oversampling_factor	[1.0, 5.0]	Amount of oversampling
DenseLoss	alpha	[0.0, 2.0]	Density weighting factor

4.4 Implementation Resources

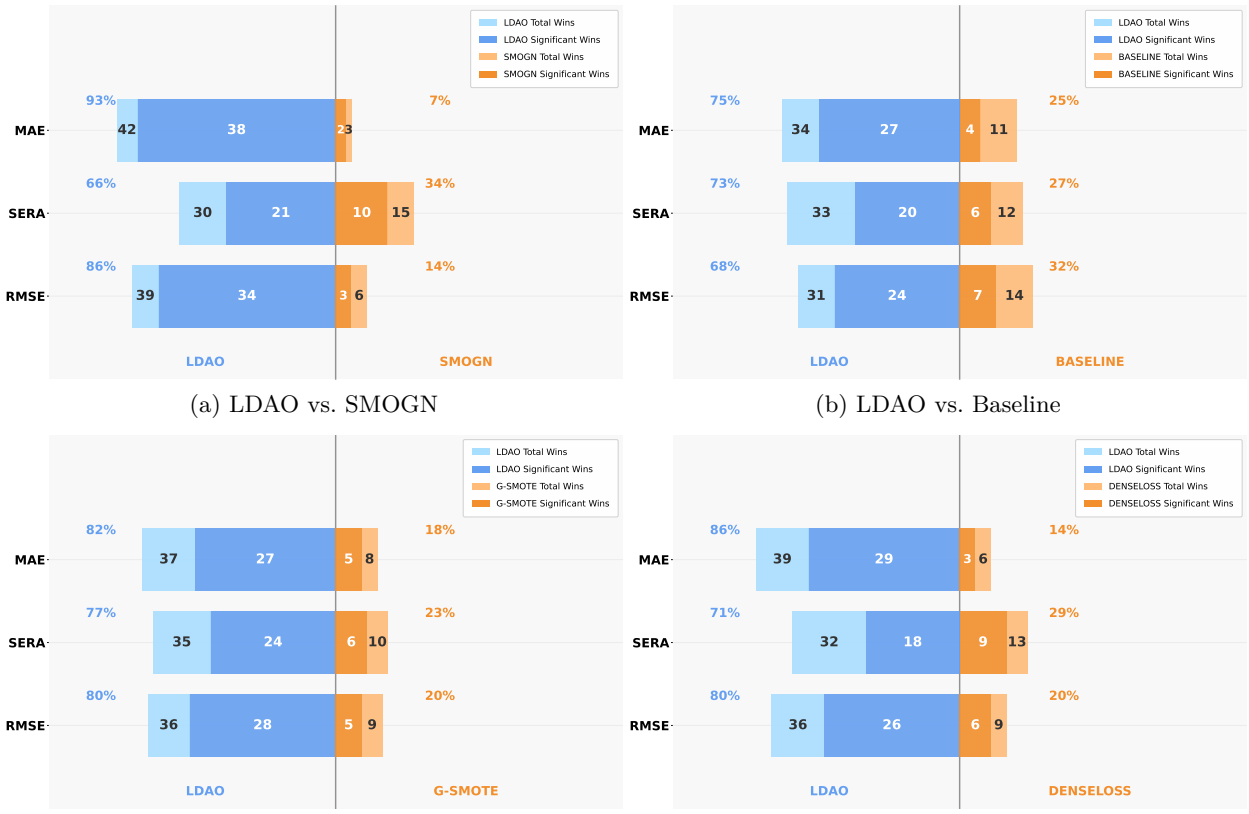
For our evaluation metrics, we utilized the SERA implementation from the ImbalancedLearningRegression Python package (Wu et al., 2022). The SMOGN method was implemented using the package developed by

Kunz (Kunz, 2020). We implemented the DenseLoss and G-SMOTE methods based on their original papers, carefully following the authors' descriptions and guidelines to ensure faithful reproduction of their approaches. To optimize the hyperparameters for each method, we employed the Optuna framework (Akiba et al., 2019). Optuna leverages Bayesian optimization using the Tree-structured Parzen Estimator (TPE) sampler, which efficiently balances exploration and exploitation during the search process. By tuning hyperparameters separately for each dataset, our approach accommodates the unique statistical properties and distribution characteristics inherent to each dataset, ensuring optimal performance for all methods.

4.5 Experimental Framework

We employed 5 runs of 5-fold cross-validation for all experiments. This outer fold cross-validation divided each dataset into five equal portions, with each fold using four portions (80% of data) for training and one portion (20% of data) as the test set. Each data portion served as a test set exactly once across the five folds in each run. For hyperparameter tuning within each fold, we further divided the training data into sub-training (80%) and validation (20%) sets. We utilized Bayesian optimization with 15 trials to efficiently search the parameter space, as it generally finds better hyperparameter values than a grid search while requiring fewer evaluations.

Table 2 presents the hyperparameters and search ranges for each method. LDAO's parameters include the oversampling multiplier and KDE bandwidth. SMOGN uses neighborhood size, sampling approach, and relevance threshold. G-SMOTE involves quantile for rarity, truncation factor, deformation factor, number of neighbors, and oversampling factor. DenseLoss works with the density weighting parameter.



(a) LDAO vs. SMOGN

(b) LDAO vs. Baseline

(c) LDAO vs. G-SMOTE

(d) LDAO vs. DenseLoss

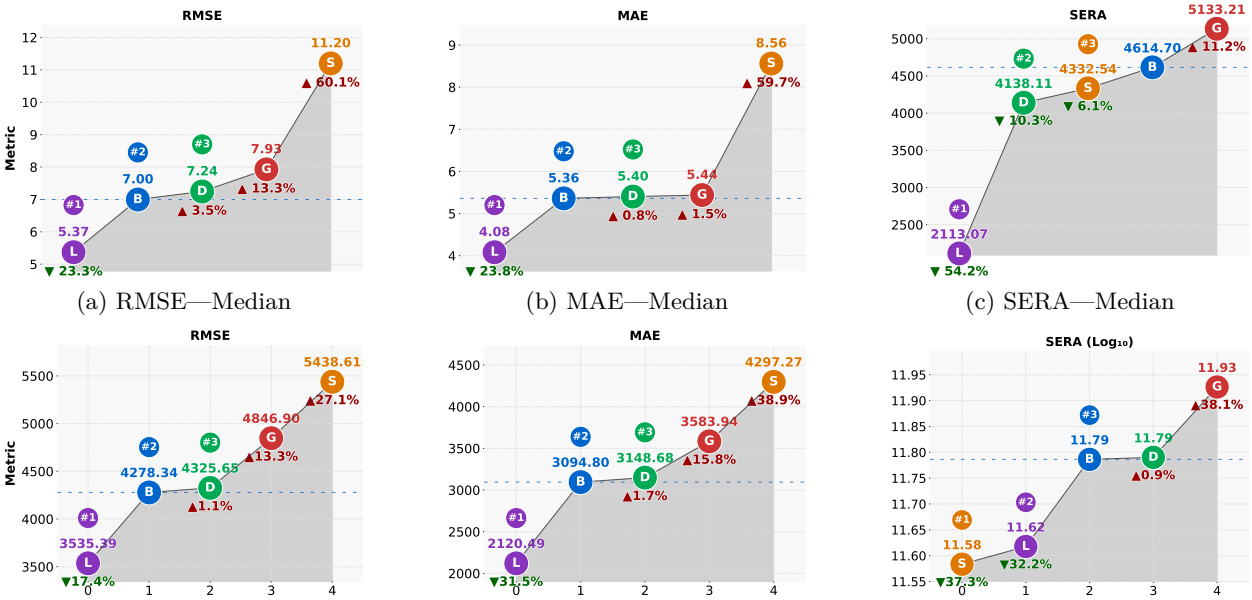
Figure 7: Pairwise comparisons of LDAO with SMOGN, Baseline, G-SMOTE, and DenseLoss across RMSE, SERA, and MAE. Light-colored bars indicate total wins, while darker segments represent statistically significant wins ($\alpha = 0.05$).

5 Results

We present the results of comparing LDAO with state-of-the-art oversampling methods across various performance dimensions. In Figure 7, we conduct pairwise comparisons for each evaluation metric (RMSE, SERA, MAE) to demonstrate how LDAO performs relative to each competing method individually. RMSE and MAE evaluate overall model performance across the entire target distribution, while SERA places greater emphasis on errors associated with rare samples.

For each dataset, a winner is determined based on the outcomes over 25 folds, meaning that each dataset contributes one win, and there are 45 wins in total. Specifically, within each dataset, the method that prevails in the majority of the 25 folds is declared the winner. This procedure is repeated against each competing method. Next, we perform the Wilcoxon Signed-Rank Test to compare LDAO's performance vector with that of the competing method. In this test, we compute the differences between the 25-fold loss metric values of LDAO and those of the competing method to construct a difference vector. The Wilcoxon test then examines whether these differences are statistically significant, using an alpha level of 0.05, which indicates a 5% risk of incorrectly concluding that a difference exists when there is none (Wilcoxon, 1945).

In Figure 7, the lighter color represents the total number of wins for each method, while the darker portion indicates the number of wins that are statistically significant. The figure illustrates that LDAO outperforms all competing methods across all evaluated metrics. Although SMOGN exhibits lower performance in terms of RMSE and MAE, its SERA measurement shows significantly superior performance in more tests, aligning with its emphasis on rare samples; nevertheless, SMOGN remains inferior to LDAO overall. The other methods display a more consistent performance across the different metrics.



(a) RMSE—Median (b) MAE—Median (c) SERA—Median

(d) RMSE—Mean (e) MAE—Mean (f) SERA—Mean

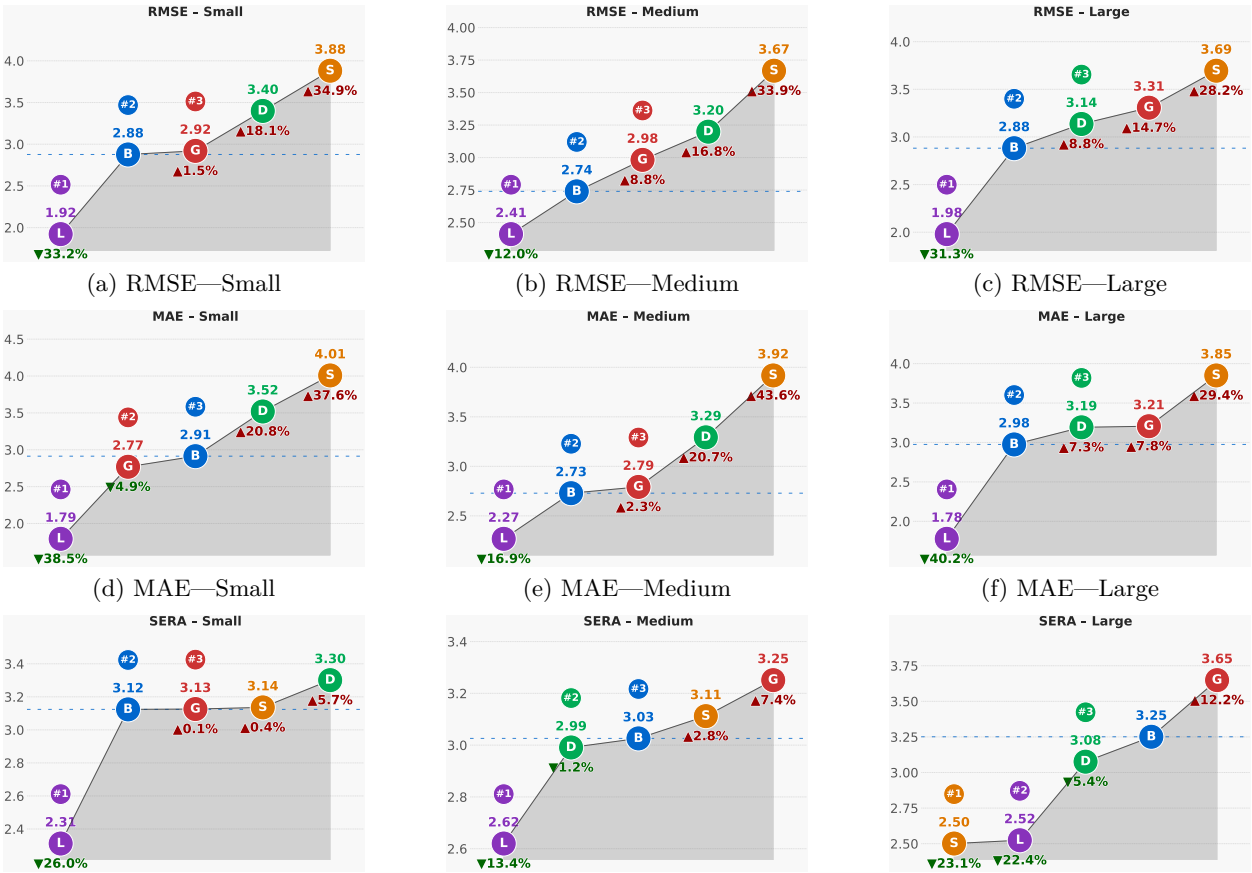
Figure 8: Comparison of methods across median and mean values of RMSE, MAE, and SERA.

In Figure 8, we computed both the median and the mean of various metrics across all datasets for each method. In this analysis, the methods were evaluated collectively rather than in pairwise comparisons. The blue circle represents the baseline benchmark, and the percentages, whether negative or positive, indicate how each oversampling technique performs relative to this baseline.

LDAO again demonstrates robust performance and surpasses its peer methods even when all techniques are assessed together. In each figure, the top three methods are highlighted, including the baseline. The only circumstance in which a peer method, SMOGN, outperformed LDAO occurred when calculating the mean SERA value; in that specific case, SMOGN achieved slightly better results, although both methods signif-

Table 3: Mean rank and counts of best (rank 1) and worst (rank 5) across 45 datasets. The lowest mean ranks are highlighted in bold black, the highest rank 1 counts in bold green, and the highest rank 5 counts in bold red.

	RMSE			SERA			MAE		
Method	Mean	R1	R5	Mean	R1	R5	Mean	R1	R5
LDAO	**2.151**	**27**	4	**2.491**	**25**	7	**2.001**	**29**	2
BASELINE	2.817	9	1	3.106	2	5	2.843	6	1
DENSELOSS	3.257	3	12	3.118	4	10	3.353	0	11
G-SMOTE	3.025	5	5	3.286	1	**14**	2.867	7	2
SMOGN	3.749	1	**22**	2.998	10	9	3.936	1	**28**
Total	–	**45**	**44**	–	**42**	**45**	–	**43**	**44**



(a) RMSE—Small (b) RMSE—Medium (c) RMSE—Large

(d) MAE—Small (e) MAE—Medium (f) MAE—Large

(g) SERA—Small (h) SERA—Medium (i) SERA—Large

Figure 9: Rank-visualizations for each metric and dataset-size category. Letters denote methods: L=LDAO, B=Baseline, D=DENSELOSS, G=G-SMOTE, S=SMOGN.

icantly outperformed the others. For the SERA mean computation across datasets, a base-10 logarithmic scale was employed to accommodate the high loss values observed on rare samples in some datasets, thereby enhancing clarity.

Table 3 presents the mean rank of each method for RMSE, SERA, and MAE across all 45 datasets. A Rank is the position given to a method when methods are sorted by performance for a given dataset and metric. Rank 1 for the lowest error (best) and Rank 5 for the highest error (worst). For each dataset and metric, a method's rank was first averaged over 25 cross-validation folds; these per-dataset ranks were then averaged across all datasets to produce the Mean column. The R1 column indicates the number of datasets in which

a method achieved the best (lowest) rank, and the R5 column indicates the number of datasets in which it achieved the worst (highest) rank. In cases where two or more methods shared virtually identical best or worst values, no count was assigned, so some column totals may sum to less than 45.

As shown in Table 3, LDAO outperforms the other methods, achieving the lowest mean ranks and the highest R1 counts across RMSE, SERA, and MAE, while SMOGN exhibits the highest R5 counts for RMSE and MAE, indicating the weakest performance on those metrics. Baseline, DenseLoss, and G-SMOTE show intermediate performance, with G-SMOTE also recording a high R5 count in SERA.

Figure 9 offers an alternative view by displaying the mean ranking of each method as a function of dataset size. The rankings separate the data sets into small, medium and large based on the classifications previously given in table 1. This evaluation aims to determine how oversampling methods perform across different dataset sizes, as some methods might be more effective on smaller datasets while others excel with larger ones.

LDAO consistently outperformed competing methods, achieving the lowest mean ranking across all metrics and dataset sizes. The only case in which a competing method performed slightly better was for the SERA metric on large datasets, where SMOGN outperformed LDAO; however, LDAO remained superior for both small and medium datasets. In this figure, the blue circle represents the baseline model benchmark, providing a reference for the performance of the various methods.

6 Discussion

An interesting observation is the strong performance of the baseline model relative to the oversampling methods. This result suggests that oversampling techniques may underperform without meticulous fine-tuning informed by domain knowledge and underscores the inherent complexity of the imbalanced regression problem compared to classification. In many continuous oversampling approaches, samples are categorized as either rare or frequent; although such a biased assumption can sometimes assist the learning process, it may also lead to diminished performance compared to the baseline. A key attribute of LDAO is its completely data-driven nature. Despite having a few parameters to adjust, LDAO does not require extensive domain expertise, making it readily applicable to a wide range of imbalanced datasets.

Oversampling approaches have the benefit of increasing representation in sparse regions. By generating synthetic samples, these methods can help models learn better from rare target values and improve performance where data is limited. The process involves creating additional data points in underrepresented areas, allowing machine learning models to develop a more complete understanding of the entire target value range. For regression tasks with imbalanced distributions, these techniques provide a practical solution to improve prediction accuracy across the entire spectrum of outcomes.

Specifically for LDAO, the approach leverages local clustering and density estimation to generate synthetic samples that more closely match the underlying data structure. LDAO aims to enhance model performance in regions where simpler approaches might struggle due to data scarcity. The method's focus on local density awareness makes it particularly useful for complex datasets with varying degrees of sparsity. LDAO especially examine sparsity in the joint distribution of features (X) and target values (y), providing a more detailed and thorough understanding of where data is truly scarce in the multidimensional space.

Despite these advantages, oversampling methods encounter several limitations. They risk overfitting when the synthetic points are too similar to existing samples and may introduce noise if they fail to accurately capture the underlying distribution. Moreover, these techniques typically require additional parameter tuning, such as selecting appropriate rarity thresholds and determining suitable oversampling factors. In LDAO's case, additional considerations include choosing the clustering parameters, specifying the number of synthetic samples, and configuring the density estimation settings. Without proper calibration, synthetic samples might overfit local patterns or miss critical variations, thereby diminishing the overall effectiveness of the approach when the underlying data characteristics or sparsity patterns are not well captured by the chosen method.

7 Conclusion

In this work, we proposed LDAO, a local distribution-based adaptive oversampling method specifically designed to address the challenges of imbalanced regression. By modeling data in a joint feature–target space and generating synthetic samples independently within identified clusters, LDAO effectively preserves the original dataset's statistical structure. Through comprehensive evaluation on a wide variety of datasets, we show that LDAO achieves strong and consistent predictive accuracy, frequently surpassing leading data-level and algorithm-level approaches, particularly where data are sparse.

Oversampling methods, in general, have proven effective at handling imbalance by improving model representation in sparse areas, yet each method has its unique strengths and applications. LDAO contributes to this field by eliminating the need for predefined rarity thresholds and undersampling, thereby preserving valuable information and offering adaptive, data-driven augmentation. As shown in our experiments, these attributes enable LDAO to maintain overall predictive accuracy while enhancing performance in challenging, underrepresented regions. Future work should continue to refine adaptive density-based sampling methods, particularly for datasets with complex, multimodal distributions.

References

Takuya Akiba, Shuji Sano, Takeru Yanase, Tetsuya Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 2019.

Jesús Alcalá-Fdez, Alberto Fernandez, Julián Luengo, Joaquín Derrac, Salvador García, Luis Sánchez, and Francisco Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2–3):255–287, 2011.

Jelena Aleksic and Miguel García-Remesal. A selective under-sampling (sus) method for imbalanced regression. *Journal of Artificial Intelligence Research*, 82:111–136, 2025.

Jusci G. Avelino, George D. C. Cavalcanti, and Renato M. O. Cruz. Resampling strategies for imbalanced regression: A survey and empirical analysis. *Artificial Intelligence Review*, 57:Article 82, 2024.

Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modeling under imbalanced distributions. *ACM Computing Surveys*, 49(2):Article 31, 2016.

Paula Branco, Luís Torgo, and Rita P. Ribeiro. Smogn: A pre-processing approach for imbalanced regression. In *Proceedings of Machine Learning Research: LIDTA*, volume 74, pp. 36–50, 2017.

Paula Branco, Luís Torgo, and Rita P. Ribeiro. Pre-processing approaches for imbalanced distributions in regression. *Neurocomputing*, 343:76–99, 2019.

Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Chumphonk Bunkhumpornpat, Krerk Sinapiromsaran, and Chumpolphon Lursinsap. Dbsmote: Density-based synthetic minority over-sampling technique. *Applied Intelligence*, 36(3):664–684, 2012.

Luis Camacho and Fernando Bacao. Wsmoter: A novel approach for imbalanced regression. *Applied Intelligence*, 54:8789–8799, 2024.

Luis Camacho, Georgios Douzas, and Fernando Bacao. Geometric smote for regression. *Expert Systems with Applications*, 193:116387, 2022.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and William Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kolcz. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 155–164, 1999.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 973–978, 2001.

Haixiang Guo, Yijing Li, Jennifer Shang, Mingyun Gu, Yuanyue Huang, and Bing Gong. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *Proceedings of ICIC 2005*, pp. 878–887. Springer, 2005.

David Harrison and Daniel L. Rubinfeld. Hedonic prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

Haibo He and Yunqian Ma. *Imbalanced Learning: Foundations, Algorithms, and Applications*. John Wiley & Sons, 2013.

Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the International Joint Conference on Neural Networks*, pp. 1322–1328, 2008.

Terrance R. Hoens and Nitesh V. Chawla. Imbalanced datasets: From sampling to classifiers. In *Imbalanced Learning: Foundations, Algorithms, and Applications*, pp. 43–59. Wiley, 2013.

Abiodun Motunrayo Ikotun, Absalom Emmanuel Ezugwu, Laith Abualigah, Bassam Abuhaija, and Heming Jia. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, 2023. doi: 10.1016/j.ins.2022.11.139.

Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019.

Amandeep Kaur and Maysam Sarmadi. Comparative analysis of machine learning techniques for imbalanced genetic data. *Annals of Data Science*, 2024.

Bartosz Krawczyk. Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

Niklas Kunz. Smogn: Synthetic minority over-sampling technique for regression with gaussian noise. PyPI, Version 0.1.2, 2020.

Xiao-Yong Liu, Junhua Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

Nuno Moniz, Luís Torgo, and Carlos Soares. Smoteboost for regression: Improving the prediction of extreme values. In *Proceedings of the 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 127–136, 2018.

Rahmat Nainggolan, Ritha Perangin-Angin, Enda Simarmata, and Achmad Tarigan. Improved the performance of the K-means cluster using the sum of squared error optimized by the elbow method. In *Journal of Physics: Conference Series*, volume 1361, pp. 012015, 2019. doi: 10.1088/1742-6596/1361/1/012015.

Mengye Ren, Weifeng Luo, and Raquel Urtasun. Balanced MSE for imbalanced visual regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 418–427, 2022.

Rita P. Ribeiro and Nuno Moniz. Imbalanced regression and extreme value prediction. *Machine Learning*, 109(9–10):1803–1835, 2020.

Rita Patrícia Almeida Ribeiro. *Utility-Based Regression.* PhD thesis, Faculty of Sciences, University of Porto, Porto, 2011.

Dave Scheepens, Ines Schicker, Klára Hlaváčková-Schindler, and Christian Plant. Adapting a deep convolutional rnn model with imbalanced regression loss for improved spatio-temporal forecasting of extreme wind speed events in the short to medium range. *Geoscientific Model Development*, 2023.

David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, New Jersey, 2nd edition, 2015.

Simon J. Sheather and Michael C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):683–690, 1991.

Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis.* Monographs on Statistics and Applied Probability. Chapman and Hall, London, 1986.

Markus Steininger, Kevin Kobs, Pascal Davidson, Andreas Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Machine Learning*, 110(8):2187–2210, 2021.

Steffen Stocksieker, Didier Pommeret, and Anne Charpentier. Data augmentation for imbalanced regression. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Steffen Stocksieker, Didier Pommeret, and Anne Charpentier. Generalized oversampling for learning from imbalanced datasets and associated theory: Application in regression. *Transactions on Machine Learning Research*, 6, 2024.

Catherine A. Sugar and Gareth M. James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003.

Muhammad Syakur, Budi Khotimah, Elly Rochman, and Bakhtiar Satoto. Integration K-means clustering method and elbow method for identification of the best customer profile cluster. In *IOP Conference Series: Materials Science and Engineering*, volume 336, pp. 012017, 2018. doi: 10.1088/1757-899x/336/1/012017.

Luís Torgo and Rita P. Ribeiro. Utility-based regression. In *Proceedings of PKDD 2007*, pp. 597–604. Springer, 2007.

Luís Torgo, Rita P. Ribeiro, José Pedro da Costa, and Susmita Pal. Smote for regression. In *Intelligent Data Engineering and Automated Learning (IDEAL 2013). Lecture Notes in Computer Science*, volume 8206, pp. 378–387, 2013.

Luís Torgo, Paula Branco, Rita P. Ribeiro, and Bernhard Pfahringer. Resampling strategies for regression. *Expert Systems*, 32(3):465–476, 2015.

Xiaoxiang Wang, Jianfeng Xu, Ting Zeng, and Le Jing. Local distribution-based adaptive minority oversampling for imbalanced data classification. *Neurocomputing*, 2020. doi: 10.1016/j.neucom.2020.05.030.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

Wenzhuo Wu, Niklas Kunz, and Paula Branco. Imbalancedlearningregression — a python package to tackle the imbalanced regression problem. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 645–648, 2022.

Zhiran Xu, Chao Zhao, Charles D. Scales, Ricardo Henao, and Benjamin A. Goldstein. Predicting in-hospital length of stay: A two-stage modeling approach to account for highly skewed data. *BMC Medical Informatics and Decision Making*, 22:Article 110, 2022.

Jing Yang, Li Xie, Qi Yu, Xiao He, and Jie Liu. Delving into deep imbalanced regression. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 8437–8447, 2021.

Xin Zhang, Maxwell L. King, and Robert J. Hyndman. A bayesian approach to bandwidth selection for multivariate kernel density estimation. *Computational Statistics & Data Analysis*, 50(11):3009–3031, 2006.

Zhi-Hua Zhou and Xiao-Yong Liu. On multi-class cost-sensitive learning. *Computational Intelligence*, 26(3): 232–257, 2010.