POLITECNICO DI TORINO
Repository ISTITUZIONALE

Differentiable Synthetic Dataset Generation for Non-Trivial Regression Tasks

Original
Differentiable Synthetic Dataset Generation for Non-Trivial Regression Tasks / Giobergia, F., Savelli, C.. - 4192:(2026). (10th International Workshop on Data Analytics Solutions for Real-Life Applications (DARLI-AP 2026) Tampere (FIN) March 2026).

Availability:
This version is available at: 11583/3015355 since: 2026-09-10T17:46:27Z

Publisher:
CEUR

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

05 November 2026

Differentiable Synthetic Dataset Generation for Non-Trivial Regression Tasks

Flavio Giobergia[1,*], Claudio Savelli[1]

[1]Politecnico di Torino, Turin, Italy

Abstract

We propose a novel method for generating synthetic regression datasets aimed at educational and evaluative settings. Unlike standard synthetic data generation approaches, which sample inputs from a predefined distribution and compute targets via a fixed function, our method optimizes the input data directly. Given a fixed target vector and a randomly initialized, frozen nonlinear model, we perform gradient-based optimization over the input features to match the targets. To avoid trivial solutions, we introduce an additional loss term that explicitly penalizes the performance of a naive baseline model, such as linear regression. The resulting datasets are guaranteed to exhibit nonlinear structure while remaining controllable, reproducible, and interpretable. We further show how to project the optimized continuous inputs into mixed-type feature spaces, including numerical, ordinal, and categorical variables. Experimental results demonstrate that the proposed approach produces datasets that are solvable by nonlinear models but systematically challenging for linear ones, making them particularly suitable for educational purposes.

1. Introduction

Synthetic datasets are widely used in the teaching of regression and machine learning, particularly in examinations and practical assignments. Instructors often require small, previously unseen datasets that can be solved within limited time constraints while still exhibiting nontrivial structure. However, commonly adopted generation strategies – such as sampling from linear or mildly nonlinear functions with noise – tend to yield problems that are either trivial for basic models or insufficiently challenging.

More advanced synthetic data generation techniques, including generative adversarial networks [1, 2, 3] and variational autoencoders [4], aim to reproduce the statistical properties of real-world data [5]. While synthetic data generation has gained increasing attention for its applications in benchmarking [6, 7], privacy preservation [8, 9], and data augmentation [10, 11], these methods are poorly suited for educational use: they offer limited interpretability, provide little control over the relative performance of different model classes, and often produce datasets whose structure is difficult for students to reason about.

In this work, we adopt a complementary perspective. Rather than fixing the input features and generating targets via a predefined mapping, we fix both the target values and the underlying nonlinear process and optimize the input features so that the desired targets are obtained when passed through the process. Input values are initialized randomly and iteratively refined via gradient-based optimization. To avoid trivial solutions, we introduce an auxiliary loss term that explicitly penalizes the performance of a naive baseline model, such as linear regression.

Finally, we transform the resulting numerical dataset into one that includes ordinal and categorical attributes. This step serves a dual purpose: it introduces additional structured information loss, leading to unexplained variance that does not stem from arbitrary noise injection, and it increases the realism and complexity of the resulting tabular data.

The proposed procedure produces regression datasets that:

1. exhibit a clear but nonlinear relationship between inputs and targets;

Published in the Proceedings of the Workshops of the EDBT/ICDT 2026 Joint Conference (March 24-27, 2026), Tampere, Finland
*Corresponding author.
flavio.giobergia@polito.it (F. Giobergia); claudio.savelli@polito.it (C. Savelli)
0000-0001-8806-7979 (F. Giobergia); 0000-0002-0877-7063 (C. Savelli)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-04-09

CEUR-WS.org/Vol-4192/DARLIAP-paper16.pdf

2. are intentionally challenging for naive baseline models;
3. include mixed feature types commonly encountered in applied settings;
4. are reproducible and controllable in terms of size and difficulty.

Overall, we propose a differentiable dataset-generation framework that directly optimizes input features under pedagogically motivated constraints, yielding learnable yet nontrivial datasets.

2. Methodology

The objective of the proposed methodology is to generate a dataset

$$\{(\mathbf{x}_i, y_i)\}_{i=1}^{N},$$

where N is a user-defined dataset size, $\mathbf{x}_i \in \mathscr{D}$ denotes a vector of independent variables, and $y_i \in \mathbb{R}$ is a continuous target value. While a variety of synthetic data generation strategies exist, we impose a set of constraints motivated by pedagogical considerations:

1. *Existence of an underlying data-generating process.* We assume the existence of a function $f(\cdot)$ such that $y_i = f(\mathbf{x}_i)$. This ensures that the regression task is well-defined and, in principle, solvable.
2. *Non-triviality of the generating process.* Simple linear or weakly nonlinear relationships often yield datasets that are trivially solved by basic models. We instead require a nonlinear process that cannot be adequately approximated by naive baselines.
3. *Presence of unexplained variance.* In real-world datasets, perfect prediction is rarely achievable due to latent or unobserved variables. Rather than injecting artificial noise, we aim to induce unexplained variance by removing information, thereby reflecting missing or inaccessible features.
4. *Mixed-type tabular structure.* The generated datasets should include numerical, ordinal, and categorical attributes, reflecting the structure of many applied tabular learning problems.

3. Generation of Non-Trivial Datasets

To address the first three desiderata, we define a nonlinear generation process f_θ, whose parameters are randomly initialized and kept fixed throughout the procedure. We also fix a vector of target values

$$\mathbf{y} = \{y_1, \dots, y_N\},$$

where each y_i is sampled from a user-specified distribution (e.g., Gaussian or uniform).

Input features $\mathbf{x}_i \in \mathbb{R}^d$ are initialized randomly and treated as learnable variables. For convenience, we denote by $\mathbf{X}$ tha matrix stacking all $\mathbf{x}_i$ row-wise. These learnable variables are optimized via gradient descent to minimize the discrepancy between $f_\theta(\mathbf{x}_i)$ and y_i, while keeping both f_θ and $\mathbf{y}$ fixed. The primary loss function is defined as:

$$\mathscr{L}_{\text{gen}} = \frac{1}{N} \|f_\theta(\mathbf{X}) - \mathbf{y}\|_2 \,. \tag{1}$$

Optimizing this loss alone may lead to degenerate solutions in which simple linear models achieve competitive performance. To counteract this effect, we introduce an auxiliary loss term that penalizes the performance of a naive baseline regressor.

Let $\boldsymbol{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ denote the closed-form solution of a linear regression model trained on $\mathbf{X}$ and $\mathbf{y}$. The corresponding mean squared error is:

$$\mathscr{L}_{\text{lin}} = \frac{1}{N} \|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|_2 \,. \tag{2}$$

The final optimization objective is:

$$\mathscr{L} = \mathscr{L}_{\text{gen}} - \lambda \mathscr{L}_{\text{lin}}, \tag{3}$$

where $\lambda \geq 0$ controls the extent to which linear solutions are penalized. Larger values of λ yield datasets that are increasingly difficult for naive models, while remaining perfectly learnable by the underlying nonlinear process.

While we focus on linear regression as the baseline model – given its canonical role in introductory regression analysis – the same framework naturally extends to polynomial or other parametric baselines via appropriate feature transformations.

3.1. Latent Variables via Feature Removal

The optimization procedure produces a dataset $\mathbf{X}$ that, by construction, allows f_θ to accurately recover the target values $\mathbf{y}$. To model the presence of latent or unobserved variables, we remove a subset of $d' \leq d$ features from $\mathbf{X}$. These features are used internally by f_θ but are not included in the final dataset, resulting in partial information loss.

We define the latent information ratio as $\rho = d'/d$. The observed dataset $\mathbf{X}'$ is obtained by applying a projection that removes the latent dimensions. This mechanism induces unexplained variance without injecting stochastic noise, reflecting missing information rather than randomness.

Through these steps, we satisfy the first three desiderata:

1. the existence of an underlying data-generating process;
2. the non-triviality of the input–output relationship;
3. the presence of unexplained variance due to latent variables.

3.2. Conversion to Ordinal and Categorical Attributes

To satisfy the final desideratum, we transform the numerical dataset $\mathbf{X}'$ into a mixed-type tabular representation. Let d_n, d_o, and d_c denote the desired number of numerical, ordinal, and categorical attributes, respectively.

We first select d_n columns of $\mathbf{X}'$ that are left unchanged, yielding the numerical attributes. Next, d_o columns are discretized using standard techniques such as equal-width binning, equal-frequency binning, or k-means discretization. Each discretized attribute may use a different number of bins, allowing for varying levels of granularity. Since the bins correspond to ordered intervals of a continuous variable, the resulting attributes are naturally ordinal.

Categorical attributes are generated by applying an arg max operation to disjoint subsets of columns. To encode a categorical attribute j with n_j possible values, we allocate n_j columns and assign each instance to the index of the column with the maximum value. This operation can be interpreted as the inverse of one-hot encoding.

To generate all d_c categorical attributes, a total of $\sum_{j=1}^{d_c} n_j$ columns is required. The total dimensionality of the initial feature space is therefore:

$$d = d' + d_n + d_o + \sum_{j=1}^{d_c} n_j. \tag{4}$$

Finally, we note that the conversion to ordinal and categorical attributes introduces additional, structured information loss that further contributes to unexplained variance. While the primary driver of latent information is the ratio ρ, these lossy transformations play a secondary role. Accurately modeling their impact is nontrivial; in practice, we recommend an empirical approach validated against baseline model performance.

3.3. Additional post-processing

The obtained dataset, after the reported steps, is characterized by numerical, ordinal and categorical features, and a target that depends on these features.

It is possible to introduce additional characteristics to the dataset to make it more useful from a pedagogical perspective. Since the focus of the proposed work is to produce a "base" dataset, the post-processing steps are left as an additional activity to be carried out as needed. Some examples are reported below.

Feature dependence. The assumption of independence of the variables is a simplifying step that makes the learning objective more straightforward. Adding dependence among the variables is possible either by introducing additional objectives (as a part Equation 3), or by introducing a post-processing step that generates combinations of existing variables.

Noise and outliers. Noise can be added to the existing data points (e.g., at the row- or column-level), to introduce decorrelation with the target variable. In addition, outlier points can be randomly generated and introduced. The fraction of noise and outliers introduced will, of course, affect the performance that can be obtained on the task.

Missing values. Similarly to noise, missing values can be introduced to enrich the pre-processing that needs to be applied to the dataset. Noise can either be added (1) at the row- or column-level, or (2) uniformly throughout the dataset, or (3) in a way that is correlated with the target feature.

4. Experiments

In this section, we present the experimental results obtained by generating datasets with the proposed method. First, in Section 4.1, we present the performance of various regression models on datasets generated using the proposed method and compare it with other baselines. Next, in Section 4.2, we show how the hyperparameters λ and ρ affect the obtained results. We present qualitative results, including low-dimensional ones, in Section 4.3. Finally, in Section 5, we present results from a real-world use case in which we employed a generated dataset as part of an exam, involving 143 participants.

4.1. Main results

We compare the datasets obtained with the proposed method with those generated using the commonly adopted scikit-learn [12] library via the `make_regression` function. This method generates random attributes by sampling them from Gaussian distributions, producing an output as a linear combination of the independent variables with additional Gaussian noise (with standard deviation 1.0). We include `make_regression` not as a strong baseline, but as a commonly used educational tool. For the proposed approach, we set $\lambda = 0.15$ and $\rho = 0$. For f_θ, we use a neural network with 5 linear layers interleaved with ReLU functions. More details on the actual implementation are available in the GitHub repository[1].

For both methods, we generate $N = 10,000$ samples and $d = 100$ attributes. For a fair comparison, we leave all the numerical features generated with our methodology. We generate 10 separate datasets with both techniques, and test them against various commonly adopted regression methods: Decision Trees and Random Forests, K-Nearest Neighbors, and Linear Regression (also with two regularization techniques: Ridge and Lasso). In all cases, we use a 70/30 train/test split to ensure a fair evaluation and evaluate the results using the R^2 score.

Table 1 shows the results. We emphasize that, unlike common situations where "higher is better", in this case we are *not* interested in achieving high performance, but rather in producing a dataset that is *interesting* from a pedagogical perspective.

With this in mind, the results are notable: first, as expected, linear regression achieves the best performance on the scikit-learn datasets, with an $R^2 \approx 1$. This occurs because their data-generation process is, by design, linear. Instead, it is clear that the proposed generation approach, by design, renders linear regression methods ineffective ($R^2 \approx 0$).

[1] https://github.com/fgiobergia/synth-datagen

Table 1
Performance comparison (mean ± standard deviation of R^2) between datasets generated with the proposed method (*our*) and `make_regression` from scikit-learn.

Model	Our method	`make_regression`
Decision Tree (depth=5)	0.600 ± 0.094	0.548 ± 0.074
Random Forest (10 models)	0.928 ± 0.014	0.805 ± 0.046
K-Nearest Neighbors (K=1)	0.934 ± 0.030	-0.185 ± 0.034
Linear Regression	-0.033 ± 0.016	1.000 ± 0.000
Lasso Regression ($\alpha = 0.1$)	0.003 ± 0.010	1.000 ± 0.000
Ridge Regression ($\alpha = 1.0$)	-0.033 ± 0.016	1.000 ± 0.000

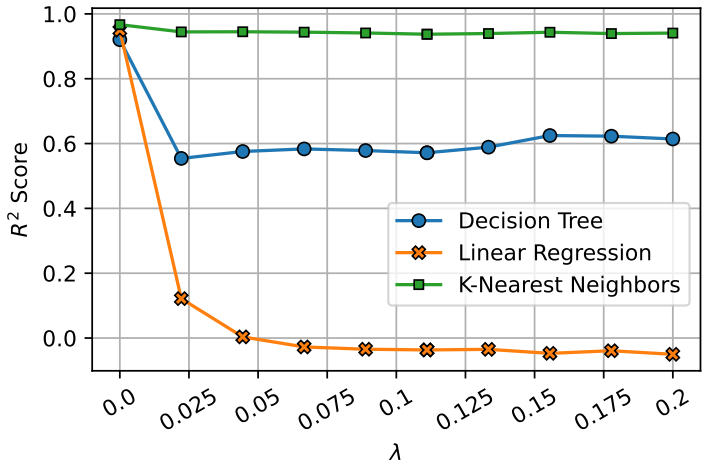


Figure 1: Effect of the baseline-penalization parameter λ on model performance. The figure reports the R^2 score of different regression models as λ increases.

Other regressors also work well for the proposed method. An additional insight emerges from the behavior of K-Nearest Neighbors regression. KNN relies on the assumption that nearby points in feature space correspond to similar target values and is therefore sensitive to the presence of meaningful local structure. On datasets generated with `make_regression`, KNN performs poorly, achieving a negative R^2, indicating that local neighborhoods in the high-dimensional feature space are largely uninformative. In contrast, KNN achieves strong performance on datasets generated with the proposed method. This suggests that the optimization-based generation procedure induces a feature space in which locality is more informative, despite the absence of an explicit locality constraint in the objective.

4.2. Influence of λ, ρ

In this section, we discuss the effect of the two main hyperparameters adopted, λ and ρ.

Figure 1 shows the performance of various regressors, as well as a linear regressor, as λ increases. As a reminder, λ serves as a parameter that governs how "difficult" the task is to solve with a linear model. We show how $\lambda = 0$ generates a very simple task that can be perfectly solved by all models. Instead, for $\lambda > 0$, the performance for the linear model is actively penalized during the training, as shown by the sharp drop in performance. Decision trees, which are known to be very simple models, are also affected by this additional complexity. Other, more robust, models are instead unaffected. This behavior can enable educators to produce datasets that work well for a given model but not for others, thereby encouraging students to explore multiple possible models for their solution.

Figure 2 instead shows the performance of the two approaches as ρ varies. We apply the same approach (i.e., incrementally remove features) to `make_regression` as well. The result is interesting:

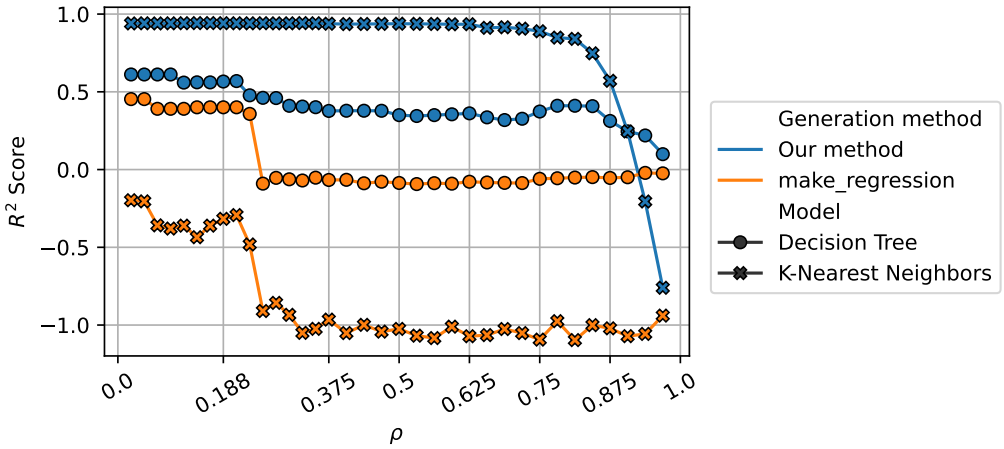


Figure 2: Impact of latent feature removal on regression performance. The figure shows the R^2 scores of the Decision Tree and K-Nearest Neighbors regressors as a function of the latent information ratio, ρ, for datasets generated using the proposed method and `make_regression`.

decision trees and KNN both have a drop in performance for large values of ρ. With the exception of very large values of ρ, however, the datasets still remain "useful" (i.e., $R^2 > 0$). Instead, for the models tested on the scikit-learn dataset, performance drops sharply for small values of ρ, rendering the models unusable ($R^2 < 0$) when partial information is removed.

4.3. Qualitative results

In Figure 3, we qualitatively compare three datasets generated with the proposed method and three generated using `make_regression`. All datasets consist of 10,000 samples and 100 features, which are projected to two dimensions using Principal Component Analysis. The PCA projections of the `make_regression` datasets consistently form compact, isotropic clusters, a behavior attributable to the Gaussian distribution of the input features. This pattern is stable across runs, indicating limited structural variability in the generated data.

In contrast, datasets produced by the proposed methodology exhibit markedly different geometric structures across runs. In all cases, the PCA projections reveal two approximately orthogonal directions of high variance, suggesting the presence of structured, non-linear relationships in the data. We attribute this behavior to the auxiliary loss term, which penalizes linear solutions and encourages nonlinearly solvable structures. To further support this interpretation, Figure 4 compares PCA projections obtained with $\lambda = 0$ and $\lambda = 0.15$. When $\lambda = 0$, the resulting dataset exhibits a more linear and homogeneous structure, whereas a positive baseline penalty yields more pronounced, structured variance.

5. Real-world Use case

To assess the practical usefulness of the proposed dataset generation methodology, we adopted the approach to generate a dataset to be used for a real-world evaluative setting. The dataset has been used as a part of an exam for a Master's course in data science, in addition to a written test, and participation to a competition-like project [13]. The task for this part of the exam consisted in addressing a regression problem, by designing and training a predictive model under time and resource constraints (90 minutes, with access to a virtual machine and Python libraries that are commonly adopted in data science).

The generated dataset contained 10 continuous, 5 ordinal, and 6 categorical attributes, and was split into 5,700 development samples (training and validation) and 1,400 test samples. Participants were provided with the development split (with target values) and evaluated on the held-out test set using the R^2 metric.

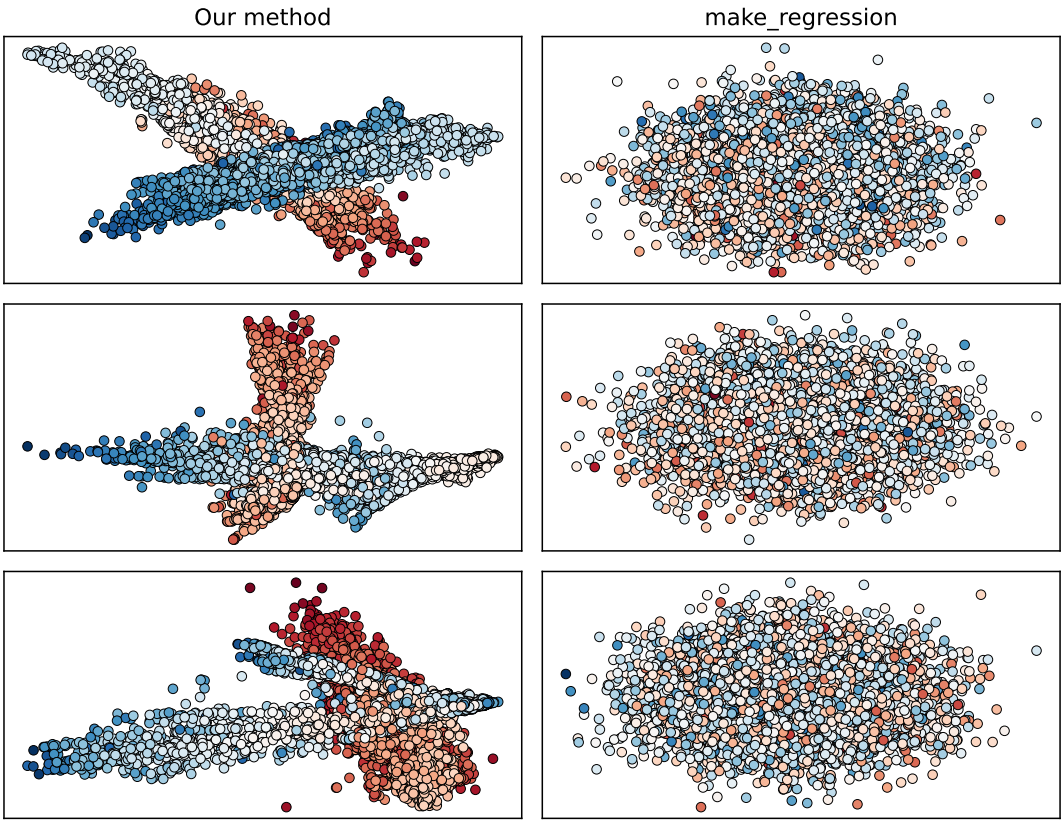


Figure 3: Qualitative comparison of generated datasets using PCA projections. Each row shows a different dataset generated with the proposed method (left) and with `make_regression` (right).

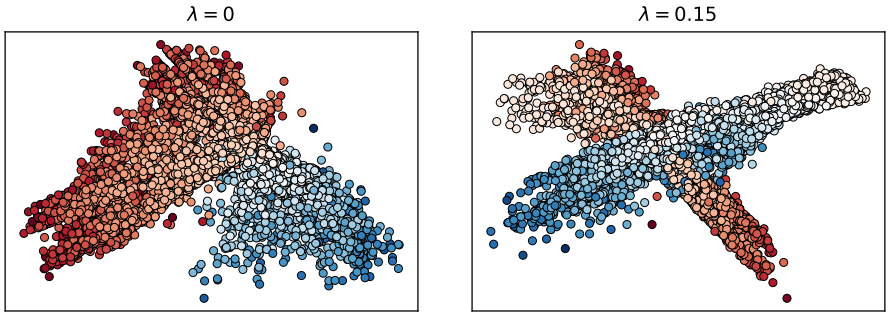


Figure 4: Effect of the baseline penalty on dataset structure. Two-dimensional PCA projections of datasets generated with the proposed method for $\lambda = 0$ (left) and $\lambda = 0.15$ (right).

A total of 143 participants took part in the exam. Of these, 52 managed to submit a working solution in the allotted time (students are allowed to submit working solutions *after* the exam, with a penalty proportional to the number of changes w.r.t. the solution submitted during the exam). Figure 5 reports the distribution of R^2 scores obtained on the test set. The left panel shows the overall performance distribution, which is highly concentrated, indicating that the task is learnable using appropriate modeling choices. At the same time, performance does not trivially saturate, as reflected by the spread of scores and the presence of a small number of low-performing solutions.

The right panel focuses on the high-performance region of the distribution and reveals that even among top-performing solutions, the achieved R^2 values exhibit meaningful variability. This result is particularly relevant in an evaluative context, as it demonstrates that the generated dataset can discriminate between solutions, rather than collapsing to the same performance.

Overall, this deployment confirms that the proposed generation method produces datasets that are

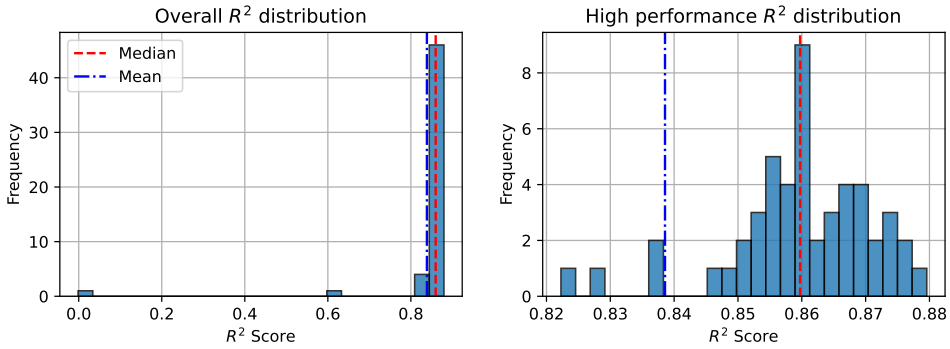


Figure 5: Distribution of R^2 scores obtained on the generated dataset. The left panel shows the overall distribution of performance, excluding extreme outliers. The right panel focuses on the high-performance region, illustrating that even among top-performing solutions, the achieved R^2 values exhibit variability.

both accessible and non-trivial in practice. The controlled nonlinearity of the task, combined with mixed feature types and latent information, yields a regression problem that admits multiple viable solutions while still rewarding careful model selection and design.

6. Discussion

The proposed method is not intended to produce realistic or privacy-preserving data. Instead, it provides a controlled mechanism for constructing datasets with known and adjustable properties. This makes it particularly suitable for educational settings, where clarity, control of difficulty, and reproducibility are more important than fidelity to real-world distributions.

Non-triviality is enforced structurally through explicit optimization objectives and information removal, rather than through arbitrary noise injection. As a result, the generated datasets remain learnable while avoiding degenerate or saturated solutions, as supported by both experimental results and real-world deployment.

We note, however, that the method has limitations. Dataset properties depend on the choice of the frozen nonlinear model, and the optimization procedure provides no theoretical guarantees of global optimality. These limitations are acceptable in the intended use cases, where controllability and practical effectiveness take precedence over formal guarantees.

7. Conclusion

We introduced a differentiable approach to synthetic dataset generation that optimizes input features to satisfy pedagogical constraints. By combining a frozen nonlinear model with an explicit baseline-penalizing objective, we generate regression datasets that are learnable yet non-trivial. The method is simple to implement, highly controllable, and well-suited for use in exams and teaching. We show the advantages of the proposed approach over existing baselines, and successfully adopted the process to generate datasets relevant to data science exams. Future extensions will mainly aim to address problems other than regression (e.g., classification, anomaly detection, and clustering). We note that each of these families of problems have different characteristics and constraints – making the extension of the current work non-obvious. We hope this work encourages further exploration of purpose-built synthetic data for educational and evaluation purposes.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA

(PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (GPT 5.2) in order to: Drafting content. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[2] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, Advances in neural information processing systems 32 (2019).

[3] L. Xu, K. Veeramachaneni, Synthesizing tabular data using generative adversarial networks, arXiv preprint arXiv:1811.11264 (2018).

[4] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[5] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, G. Kasneci, Deep neural networks and tabular data: A survey, IEEE transactions on neural networks and learning systems 35 (2022) 7499–7519.

[6] P. Orzechowski, J. H. Moore, Generative and reproducible benchmarks for comprehensive evaluation of machine learning classifiers, arXiv preprint arXiv:2107.06475 (2021).

[7] F. Giobergia, E. Pastor, L. de Alfaro, E. Baralis, A synthetic benchmark to explore limitations of localized drift detections, in: International Workshop on Discovering Drift Phenomena in Evolving Landscapes, Springer, 2024, pp. 101–110.

[8] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, J. Sun, Generating multi-label discrete patient records using generative adversarial networks, in: Machine learning for healthcare conference, PMLR, 2017, pp. 286–305.

[9] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, FAME: Fictional actors for multilingual erasure, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference, European Language Resources Association, 2026.

[10] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification, Neurocomputing 321 (2018) 321–331.

[11] F. Borra, C. Savelli, G. Rosso, A. Koudounas, F. Giobergia, Malto at semeval-2024 task 6: Leveraging synthetic data for llm hallucination detection, in: Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), 2024, pp. 1678–1684.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, the Journal of machine Learning research 12 (2011) 2825–2830.

[13] G. Attanasio, F. Giobergia, A. Pasini, F. Ventura, E. Baralis, L. Cagliero, P. Garza, D. Apiletti, T. Cerquitelli, S. Chiusano, Dsle: a smart platform for designing data science competitions, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, 2020, pp. 133–142.