Towards High Supervised Learning Utility Training Data Generation: Data Pruning and Column Reordering

Tung Sum Thomas Kwok[*]
tk1018@ucla.edu
University of California, Los Angeles
Los Angeles, California, USA

Zeyong Zhang[*]
zyong@bu.edu
Boston University
Boston, Massachusets, USA

Chi-Hua Wang[*]
chihuawang@ucla.edu
University of California, Los Angeles
Los Angeles, California, USA

Guang Cheng
guangcheng@ucla.edu
University of California, Los Angeles
Los Angeles, California, USA

Abstract

Tabular data synthesis for supervised learning ('SL') model training is gaining popularity in industries such as healthcare, finance, and retail. Despite the progress made in tabular data generators, models trained with synthetic data often underperform compared to those trained with original data. This low SL utility of synthetic data stems from class imbalance exaggeration and SL data relationship overlooked by tabular generator. To address these challenges, we draw inspirations from techniques in emerging data-centric artificial intelligence and elucidate Pruning and ReOrdering ('PRRO'), a novel pipeline that integrates data-centric techniques into tabular data synthesis. PRRO incorporates data pruning to guide the table generator towards observations with high signal-to-noise ratio, ensuring that the class distribution of synthetic data closely matches that of the original data. Besides, PRRO employs a column reordering algorithm to align the data modeling structure of generators with that of SL models. These two modules enable PRRO to optimize SL utility of synthetic data. Empirical experiments on 22 public datasets show that synthetic data generated using PRRO enhances predictive performance compared to data generated without PRRO. Specifically, synthetic replacement of original data yields an average improvement of 26.74% and up to 871.46% improvement using PRRO, while synthetic appendant to original data results with PRRO-generated data results in an average improvement of 6.13% and up to 200.32%. Furthermore, experiments on six highly imbalanced datasets show that PRRO enables the generator to produce synthetic data with a class distribution that resembles the original data more closely, achieving a similarity improvement of 43%. Through PRRO, we foster a seamless integration of data synthesis to subsequent SL prediction, promoting quality and accessible data analysis. All data and codes can be accessed through the following anonymous GitHub link: https://anonymous.4open.science/r/PRRO_utility.

CCS Concepts

• **Theory of computation** → *Program reasoning*.

Keywords

Training Data Generation, Supervised Learning Model Training, Synthetic Data Utility, Data Pruning, Column Reordering

ACM Reference Format:
Tung Sum Thomas Kwok, Zeyong Zhang, Chi-Hua Wang, and Guang Cheng. 2025. Towards High Supervised Learning Utility Training Data Generation: Data Pruning and Column Reordering. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Supervised learning ('SL') training with tabular datasets is one of the most common machine learning tasks to discover patterns, learn representations and make predictions. However, real-life SL task faces constraints in data privacy and data quality, e.g. real-life datasets may contain sensitive information [52] and insufficient historical occurrences for effective analysis [71]. Training data synthesis offers a promising solution to the above two problems by 1) replacing the original real datasets with 'fake' data of similar statistical behavior and 2) appending the original real datasets with synthetic observations over the minority classes to increase exposure of the minorities [6, 21, 28, 38, 64]. The development of the training data synthesis area has been immensely accelerated by the speedy evolution of generative AIs in terms of model training cost and performance [13, 56, 70] as well as practitioners' realization of value after applications in business, medicine and marketing industry [2, 43, 49, 54, 65].

As generative models continue to push the boundaries of data synthesis, it becomes crucial to examine not just the fidelity of the generated data, but also the utility: how effectively it can be used in downstream machine learning tasks. This evaluation metric plays a particularly important role in tabular data for conducting subsequent data analysis work, despite being underexplored compared to synthetic fidelity in many works [31, 73]. Combined with

[*] Both authors contributed equally to this research.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

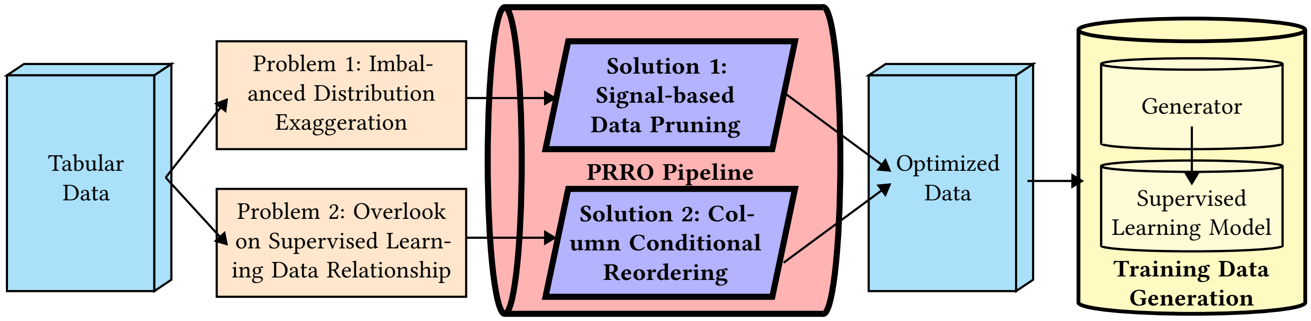

Tabular Data
Problem 1: Imbalanced Distribution Exaggeration
Problem 2: Overlook on Supervised Learning Data Relationship
Solution 1: Signal-based Data Pruning
PRRO Pipeline
Solution 2: Column Conditional Reordering
Optimized Data
Generator
Supervised Learning Model
Training Data Generation

Figure 1: Overview of the PRRO Pipeline which addresses two problems of tabular data that leads to low synthetic utility

works showing that high fidelity synthetic data does not necessarily contribute to high utility [69], there is the urgent industrial need to explore major contributing factors that govern synthetic data utility. Specifically, high SL utility of synthetic data is primarily hurdled by the following factors: (1) Class imbalance is extremized (Fig. 2) due to distribution smoothening in machine learning models [51], which treats low-occurrence class as outliers / noise [42] and exaggerating the imbalance (Table 1). When studying synthetic data of highly imbalanced dataset, which undergoes two consecutive smoothening steps in the generator and SL model, the exaggeration problem is especially severe. The urgency of mitigating this exaggeration is accelerated along the increasing prevalence of synthetic data.; (2) SL model's key focus on predictor-feature conditional relationship is overlooked by the table generator. While the latter model type regards data relationships over the entire table equally [3, 45], the former has a well-established statistical assumption of predictor conditional on features [18], which generator weighs less on. Bridging the two conditional relationship assumptions is needed to allow smooth translation between output from upstream generators and input to downstream SL models.

To address the aforementioned challenges, in this paper, we introduce a novel PRuning and ReOrdering PRRO pipeline, a pre-synthesis-processing pipeline that guides the generator to generate high SL utility synthetic data by complying to common SL techniques and assumptions. The initial phase of PRRO pipeline aims to narrow down the focus of data modeling, by emphasizing on the group of data that we seek to study and pruning observations that have low correlation to our interest group. This balances the dataset class distribution, hence reducing the extent of exaggeration during synthesis. Following this, PRRO employs a reordering module to standardize optimal order of table features for data synthesis. This guides the generator to establish a tabular conditional relationship comparable to that assumed under downstream SL models, hence preventing drop of SL utility due to overlooking of key relationships. All in all, this data pre-synthesis-processing approach bridges the difference in assumptions between generators and downstream SL models, streamlining the 'generate and predict' pipeline.

In our study, we conduct extensive experiments across two application scenarios of synthetic data, namely data replacement and appendant, using datasets from a large collection of prediction tasks in previous studies [10, 22, 59]. This collection includes 22 different types of classification tasks across identifying between age of creatures, financing behaviors of users, academic performance of students, and advertisement-clicking records of online browses. Our study stands out for its coverage on evaluating SL utility of synthetic data under different applications and class distribution variability, setting a new benchmark for SL utility evaluation when compared to previous work [3, 39, 60]. The empirical results highlight improvement in overall SL utility when original data is appended with synthetic data generated through the pipeline. To summarize, our main contributions are as follows:

(1) In Sec. 2, formally defining supervised learning utility and hence synthetic data utility for synthetic data evaluation specifically;
(2) In Sec. 3, identifying two major contributing factors to low SL utility of synthetic data, namely generator exaggerating class imbalance and overlooking key SL data relationship, hence introducing the PRRO pre-processing pipeline, which prunes and reorders tabular data to address the two factors;
(3) In Sec. 4, showing PRRO implementation contributes to consistent SL utility improvement over experiments on a broad range of datasets.

2 Preliminary

Supervised Learning Utility of Synthetic Training Data

Given a labeled training dataset $D_T = \left\{\mathbf{x}_{T,i}, y_{T,i}\right\}_{i=1}^{n_T}$, where $\mathbf{x}_{T,i}$ denotes the features for observation i and $y_{T,i}$ the corresponding label, our goal is to train a supervised learning (SL) model h to predict y from $\mathbf{x}$. Formally, we seek a model $h(y \mid \mathbf{x})$ that minimizes the expected loss over D_T :

$$l\left(h; D_T\right) = \frac{1}{n_T} \sum_{i=1}^{n_T} l\left(h; \mathbf{x}_{T,i}, y_{T,i}\right) \tag{1}$$

Given a validation dataset $D_V = \left\{\mathbf{x}_{V,i}, \boldsymbol{y}_{V,i}\right\}_{i=1}^{n_V}$, the performance of a supervised learning model h is assessed via the **validation loss**,

$$l\left(h; D_V\right) = \frac{1}{n_V} \sum_{i=1}^{n_V} l\left(h; \mathbf{x}_{V,i}, y_{V,i}\right) \tag{2}$$

In many classification settings, for instance, y may be a binary class label (0 or 1), and a common choice for $l(\cdot)$ is the cross-entropy loss: $l(h;\mathbf{x},y) = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$ where $\hat{\boldsymbol{y}} = h(\mathbf{x})$ is the predicted probability of the positive class. The validation loss then captures the average divergence between the predicted and true labels on the validation set.

Definition 2.1 (Supervised Learning Utility). Supervised learning utility is defined as the difference in validation loss between a *target model* h_A and a *baseline model* h_B on the same validation dataset D_V :

$$U\left(h_A; h_B, D_V\right) = l\left(h_A; D_V\right) - l\left(h_B; D_V\right) \tag{3}$$

Definition 2.1 can be tailored to measure *synthetic data utility* by comparing a model trained on synthetic data to one trained on the original data. Specifically, let D_T be partitioned into a *generator-training* set $D_{ST} = \left\{\left(\mathbf{x}_{ST,i}, y_{ST,i}\right)\right\}_{i=1}^{k_{ST}}$ and a *holdout* set $D_H = \left\{\left(\mathbf{x}_{H,i}, y_{H,i}\right)\right\}_{i=1}^{k_H}$ to prevent the generator from over-fitting all training data points [14, 50]. A data generator $\mathcal{G}_A$ is then fitted using D_{ST} to produce a synthetic dataset $\tilde{D}_A$, i.e.

$$\mathcal{G}_A : D_{ST} \longrightarrow \tilde{D}_A \tag{4}$$

Afterward, one can train separate predictive models h_D on either the original dataset D_T or the synthetic dataset $\tilde{D}_A$.

Definition 2.2 (Synthetic Data Utility). Synthetic data utility is the supervised learning utility (Definition 2.1) when the target model is trained on synthetic data $\tilde{D}_A$ and the baseline model is trained on the original data D_T :

$$U\left(h_{\tilde{D}_A}; h_{D_T}, D_V\right) = l\left(h_{\tilde{D}_A}; D_V\right) - l\left(h_{D_T}; D_V\right) \tag{5}$$

This measure captures how well the synthetic dataset $\tilde{D}_A$ *replaces* the original data D_T. To measure how synthetic data may *augment* the original data, one can define a combined dataset $\hat{D}_A = D_T \cup \tilde{D}_A$. The synthetic data utility in this appendant scenario is then

$$U\left(h_{\hat{D}_A}; h_{D_T}, D_V\right) = l\left(h_{\hat{D}_A}; D_V\right) - l\left(h_{D_T}; D_V\right). \tag{6}$$

For a concrete example in a classification task, this utility would reflect how adding (or replacing with) synthetic data affects metrics such as validation accuracy or cross-entropy loss, indicating whether the synthetic dataset enhances or degrades the predictive performance of the trained model.

3 Proposed Approach - Pruning and ReOrdering PRRO Pipeline

We propose the Pruning and ReOrdering (PRRO) pre-processing pipeline, to address two factors observed that lead to degradation of SL utility for synthetic data: 1) Imbalanced class distribution is leveraged by generators, and 2) SL data relationship is overlooked in existing synthetic data research, which misguides generator from capturing the correct column relationship. The PRRO pipeline modifies the original dataset into a format D^* that bridges the gap between generators and SL models for data synthesis. After synthesis, the resulting dataset is in a post-pipeline format and must be inversely transformed to its original structure for consistency.

3.1 Signal-based Data Pruning to mitigate imbalance class exaggeration

Generators, like most machine learning models, exaggerate on imbalanced dataset and heavily favor the majority class due to distribution smoothening. Signals from minority class of interest are dominated by the majority class which contains a lower signal-to-noise ratio. We observe generators generating *only* majority class data points in some highly imbalanced datasets (Table 1), indicating a more extreme class distribution in synthetic data over original data. With the synthetic class distribution being more imbalanced, SL models struggle further in capturing signals from the minority class, rendering low SL utility for the dataset. Techniques such as oversampling and SMOTE [7, 61] are proposed to balance the imbalanced distribution, but [26] shows that oversampling methods are unreliable to learn from class imbalanced data because of their tendency to generate the 'majority of minority', instead of considering the distribution of minority class, e.g. SMOTE synthesizes data by interpolating existing minority group, but interpolation struggles in high-dimensional spaces [16] when the distance between points becomes less meaningful due to "curse of dimensionality" [47]. In turn, works on undersampling, specifically data pruning such as DRoP carry analogies with distributionally robust optimization methods to address class/group bias [66]. However, while the DRoP method is respected to take into account class-conditional independence, it does not hold strongly for more sophisticated group-wise datasets when the correlation across different observations is different. Therefore, we propose a data pruning method (Sec. 3.1) that further considers different class-conditional relationships across observations, hence focusing on data having high correlation with our group of interest.

To address the exaggeration of imbalanced dataset, we propose the signal-based data pruning module to balance the class distribution of original data to mitigate the exaggeration effect of generator. Given the training dataset D_T, we extract the subgroup containing observations labeled with our interested class (denoted as $D' \subset D$) and leave the remaining non-interested class observations (denoted as $\not D' \subset D$). By iteratively going through each data point in the non-interested class subgroup, we identify and extract observations with considerable correlation with the interested subgroup data: $D'^{*} = \{d \in \not D' | \exists d' \in D' : corr(d,d') > \tau\}$. These observations with non-interested class but containing high relevancy to our study interest are then concatenated with the original interested subgroup to form the desired signal group: $D^* = D' \cup D'^*$. This signal group with higher signal-to-noise ratio serves as exemplar and guides the generator towards studying specific traits of our class of interest. [74].

This data pruning module is specifically designed to assist in data analytics tasks over imbalanced datasets. We illustrate the method with a binary classification setting, where the group of interest is the positive observations. Machine learning models tend to focus and bias towards the majority class, e.g. in the modeling of CTR datasets where the average industrial clickthrough rate is about 0.57% [5], both the synthetic and downstream predictive models are incentivized to model the majority class that accounts for 99.43%

Algorithm 1 Signal-based Data Pruning Algorithm

```
Input: The training dataset $D_T = \{(\mathbf{x}_{T,i}, y_{T,i})\}_{i=1}^{k_T}$, where $D' = \{(\mathbf{x}_{T,i}, y_{T,i}) | y_{T,i} = 1\} \subset D$ and $D'' = \{(\mathbf{x}_{T,i}, y_{T,i}) | y_{T,i} = 0\} \subset D$.
Output: The signal-containing subgroup containing interested class observations and other non-interested class observations with high correlation to the interested class observations: $D^* = D' \cup \{(\mathbf{x}_{T,i}, y_{T,i}) \in D'' | \exists x_k \in D' : \mathrm{corr}(\mathbf{x}_{T,i}, \mathbf{x}_k) > \tau\}$.
Initialize $\tau$, $D^* \leftarrow D'$;
for each $d0 \in D''$ do
    for each $d1 \in D'$ do
        if $\mathrm{corr}(d0, d1) > \tau$ then
            $D^* \leftarrow D^* \cup d$;
            break
        end if
    end for
end for
```

in terms of maximizing overall accuracy. This majority bias, which also extends to state-of-the-art models such as LLM [20], can be problematic when the model generates / predicts no positive observations at all, as the majority negative class is not of our interest, hence contrasting the research goal. The proposed data pruning method increases the density of the informative observations in the subdataset to be studied, allowing the generator to lean towards studying and generating the minority class.

Algo. 1 shows the algorithm for the Signal-based Data Pruning module in binary classification case. The algorithm firstly separates the dataset into the group of interested and non-interested subgroups. Each observation from the non-interested class will undergo a correlation measurement with every observation from the interested class, and will be classified as the signal-containing subgroup if this observation exhibits significant correlation with any interested class observations. The correlation measurement in our experiment involves a numerical measurement using the Spearman correlation, whose motivation will be further discussed in Sec. 4.1. The algorithm focuses on pruning the observations that 1) contain a class that researchers are not interested in learning, and 2) have negligible contribution in assisting researchers to learn about the class of interest, which summarizes into noisy observations with limited contribution to the SL task.

3.2 Column Conditional ReOrdering to replicate supervised learning data relationship

Generators often overlook critical data relationships required by SL models. This is because SL models focus specifically on the predictor-features conditional relationship, whereas other machine learning paradigms, such as unsupervised learning and reinforcement learning, examine broader relationships across features and environmental states [63]. While synthesizers typically prioritize the overall tabular structure to meet the needs of various machine learning tasks, they tend to neglect the specific predictor-feature conditional relationship that is vital for supervised learning models. As a result, synthetic data does not sufficiently exhibit the SL-focused relationship. Generators study the inter-column conditional relationships either by the overall table cross-feature relationship [29, 68] or the column-by-column sequential relationship [3, 45]. Specifically, state-of-the-art generators such as LLM are brittle to ordering of inputs, even if the ordering does not alter the underlying task [1, 9]. In contrast, SL models analyze the conditional relationship between the predictor and features while assuming minimal conditional dependence among features [18]. When such dependencies are non-negligible, additional preprocessing techniques are applied to mitigate multicollinearity and prevent overfitting [24, 57, 58]. Consequently, the generator does not adequately account for the conditional relationship between the predictor and features when generating SL-compatible data. To bridge this gap, we propose the Column Conditional Reordering method (Sec. 3.2) by replicating the predictor-features conditional relationships of SL models using generators' sequential-column-based assumption. A closer resemblance of conditional relationship modeling between the two models guides the data generator to 'think and learn' like the SL model and to generate data accordingly.

Inspired by confidence learning method that trains SL model by prioritizing observations with higher prediction confidence, i.e. more obvious patterns [41], we propose a Column Conditional Re-Ordering module to guide the generator to model data in closer resemblance as SL model. SL model has a well-established assumption on studying the conditional distribution of predictor given features. While generators are not originally designed to focus on this specific conditional relationship, LLM-based generators do model data conditionally under a sequential-column-based assumption. By rearranging the column features, the sequential-column-based assumption of generators can be leveraged to mimic the conditional relationship between predictor and features. Subsequent SL model training can be enhanced by utilizing this synthetic dataset generated under the closely-aligned conditional relationship.

We conduct the experiment by comparing between the synthetic data utility of synthetic data generated under different predictor-features conditional relationship. Our experiment does not emphasize on the relationship between features, due to the assumption that feature data for SL model training should have a negligibly low multicollinearity. Under this assumption, the conditional distribution of a feature given previous features equates to the marginal distribution of that feature itself: $f(B|A) = f(B) \Longleftrightarrow A \perp B$. Therefore, the only conditional distribution to be considered is then between the features and the predictor, where the predictor should be placed after the features to mimic a conditional relationship of predictor given the features assumed from SL models. The first batch of data is inputted into the generator accordingly: $\{\mathbf{x}_T, y_T\}$ for generation. To show the importance of assumption compliance, we also generate another batch of data by placing the predictor before the features as a control group: $\{y_T, \mathbf{x}_T\}$, simulating the synthesis under the features given predictor conditional relationship.

Algo. 2 standardizes the Column Conditional ReOrdering method

Algorithm 2 Formulation of predictor-feature conditional relationship in generator

```
Input: A training and a validation dataset: $D_{T/V} = \{(\mathbf{x}_{T/V,i}, y_{T/V,i})\}_{i=1}^{k_{T/V}}$
Output: Synthetic data utility difference when data is generated in compliance with a supervised learning model versus when it is not.
Initialize generator $\mathcal{G}_A$ predictive model $h$
$X \leftarrow \mathbf{x}_T$;
$df_1 \leftarrow \text{concat}(X, y)$;
$df_2 \leftarrow \text{concat}(y, X)$;
Train two $\mathcal{G}_A$ using $df_1$ and $df_2$ separately;
Synthesize $\tilde{df}_1, \tilde{df}_2$;
Initialize predictive models $h_1, h_2$;
Train $h_1$ and $h_2$ with $\tilde{df}_1$ and $\tilde{df}_2$ respectively;
Compare $error(h_1(\mathbf{x}_V) - y_V)$ and $error(h_2(\mathbf{x}_V) - y_V)$;
```

by formulating predictor-feature conditional relationship in generator to align with the respective data modeling focus of SL models. Regardless of where the predictor is located initially in the dataset, the algorithm ensures that the predictor would be placed at the last column of the dataset to guide the generator to model the predictor-feature conditional correlation. Given the no-multicollinearity assumption, the method does not reorder the features sequence, despite an extension of the Column Conditional ReOrdering module will be discussed in Sec. 5 which further takes into account multicollinearity problem over realistic datasets. Moreover, the algorithm adds a control group testing by arbitrarily placing the predictor column of all datasets to the first column. Both synthesis will be carried out under the same generator setting detailed in Sec. 4.1 to mitigate potential modeling variance. This guides the generator in modeling in different conditional distribution assumptions between features and predictor, hence validating the significance to have the generator complying with the requirement in SL models.

4 Experiment

We are now ready to experiment on the PRRO pipeline, by pruning observations with low signal-to-noise ratio and comparing between synthetic data utility between placing the predictor before and after the features in the dataset. The experiment is conducted over 22 binary classification datasets with different levels of class distribution.

4.1 Experimental Setups

Datasets Pre-processing. We conduct our experiments over a collection of real-world datasets, listed in Table 2, commonly used in previous works for tabular classification tasks [10, 22, 31, 59]. These datasets differ with respect to the number of observations, ranging from 96 up to 134,868 observations. There are also variations in the number of features each dataset contains, ranging from 2 to 14 numerical variables and 1 to 19 categorical variables. To pre-process the datasets, we take the following three steps: **1) Each dataset is divided into a training, holdout and validation dataset** according to Sec. 2 in a 40%:40%:20% ratio [14, 50]. This is to check if the synthetic data overfits the training dataset, as it implies excessive information leakage during data synthesis [75]; **2) We subset the CTR dataset from [12] into multiple subdatasets to conduct a more in-depth study on synthetic data utility of extremely imbalanced datasets** (with positive rate below 5%). The dataset is separated according to the types of advertisement channel and advertisement contents to mimic advertisement clicking behavior under different electronic devices and channels, so that each subdataset is considered independent and aligns with authorized marketing data guideline [27], and **3) We obtain all non-CTR datasets directly from HuggingFace API, without any additional manual pre-processing** from the team to isolate the usage of pipeline for efficacy testing. Details of the datasets are included in Appendix 2.

Pipeline Implementation. The two modules in PRRO pipeline are implemented accordingly as follows: **1) The implementation of the Signal-based Data Pruning module (Algo. 1) involves defining correlations between observations**. Since most datasets contain both continuous and categorical variables, the Spearman rank correlation coefficient is used to ensure broad applicability given its ability in handling both ordinal and continuous data [55]. In our experiment, an observation from the non-interest subgroup is considered correlated with the interest subgroup if its Spearman correlation coefficient exceeds 0.3 with any observation in the interest subgroup. This is because a Spearman correlation of 0.3 represents a weak positive monotonic relationship [11], where this weak correlation still holds useful information in real-world data in areas such as marketing and healthcare [37, 40]; **2) The Column Conditional ReOrdering module compares two batches of standardized datasets, both containing the same data**. In the first batch, the predictor is manually moved to the last column (if it is not already there), while in the second batch, the predictor is moved to the first column (if it is not already there). This approach mimics modeling the data in terms of the conditional relationship between predictors and features versus features and predictors. By adjusting the position of the predictor, we account for variations in how predictors are positioned across datasets, ensuring standardized comparisons. This results in three distinct experimental settings: (1) the model trained on the original data, (2) synthetic data following the SL modeling structure: $\tilde{D} = (\tilde{y}, \tilde{X})$, and (3) synthetic data that does not follow the SL modeling structure: $\tilde{D} = (\tilde{X}, \tilde{y})$.

Model Specifications. There are two data modeling stages on generators and classification models. All datasets are experimented under the same generator and classification models of the following specifications: **1) Regarding the data generator, this experiment utilizes the `REaLTabFormer` package from [60] as the primary generator**, with regard to LLM-based generators' outperformance from other works [3, 31, 39, 60]. While the online available `REaLTabFormer` can be installed directly, the model is initially designed for a fine-tuned but out-dated GPT-2 model. It is then rewritten to allow generalized usage of LLMs available in HuggingFace along with some bugs fixed to account for the generative power of SOTA LLMs (updated package will be released in GitHub in the future after completion in generalizing multi-table use cases). In this experiment, we utilize the lightweight Llama-3.2-1b [36] as baseline because its small size allows usage from most local computers for reproducibility. For each synthesis, the LLM is fine-tuned with 500 epochs and 10 batches per epoch; **2) Regarding the downstream SL classification models, we utilize multiple classification models to study if the pipeline is model-consistent**. The models include Logistic Regression [48], four Tree-based Classifiers: Decision Tree, Random Forest, Gradient Boosting and Extreme Gradient Boosting [4, 8, 19, 53], three Naive Bayes models: Gaussian, Complement and Multinomial Naive Bayes [15, 35, 67], and two neural network models: DNN and DeepFM [23, 32]. Aside from the neural network models that are written from PyTorch [44], with ReLU activation function and 3 layers of 256 nodes in the deep component, other models are called as default class from the Sci-kit Learn package [46].

Evaluation Metric. Evaluation of synthetic data utility is conducted on how the synthetic data serves as the replacement and appendant to the original data (Def. 2.2). Given the experiment is binary-classification focused, we evaluate the synthetic data utility based on classification accuracy. However, measurement based on overall accuracy can be problematic when measuring highly imbalanced datasets, for instance, if making all predictions to be the majority class yields a high prediction accuracy. **Therefore, we further evaluate synthetic data utility with respect to precision, recall, F1 score and AUC score**, by replacing the loss function in Def. 2.2 with the each respective metric function. We denote the metric set $\mathcal{M}$ containing the above metrics that evaluates dataset D as $\mathcal{M}_D = \{P_D, R_D, F1_D, AUC_D\}$, where

$$\begin{aligned}
P_D &= \mathbb{P}_{(\mathbf{x},y)\in D_V}(y = 1|h_D(\mathbf{x}) = 1),\\
R_D &= \mathbb{P}_{(\mathbf{x},y)\in D_V}(h_D(\mathbf{x}) = 1|y = 1),\\
F1_D &= 2 \times \frac{P_D \times R_D}{P_D + R_D},\\
AUC_D &= \mathbb{P}_{(\mathbf{x}_1,y_{V,1}),(\mathbf{x}_2,y_2)\in D_V}(h_D(\mathbf{x}_1) > h_D(\mathbf{x}_2)|y_1 = 1, y_2 = 0).
\end{aligned}$$

4.2 Results

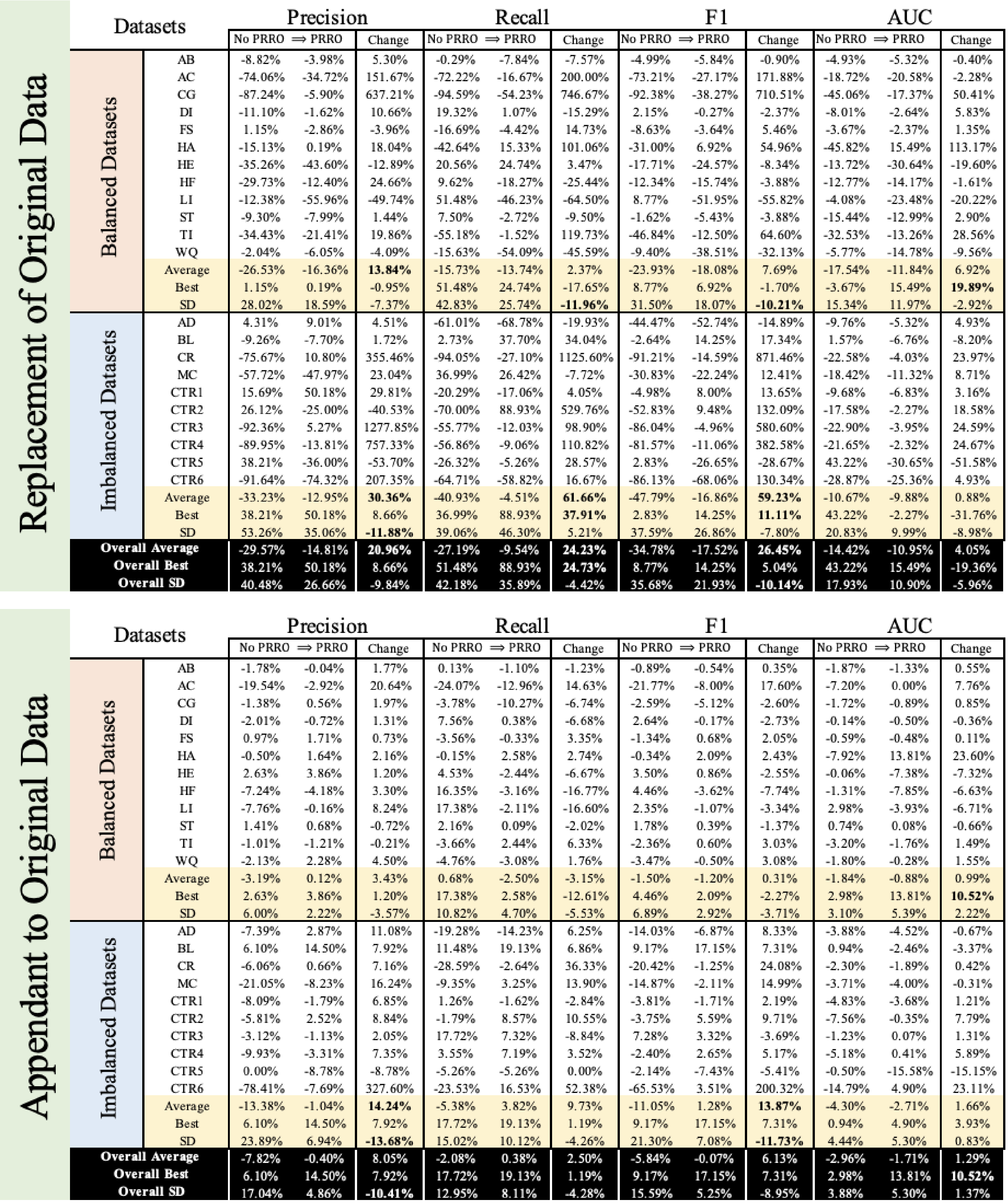
Three experimental findings, illustrated by Table 1 and Fig. 2, highlight a consistent improvement of synthetic data utility after implementing the PRRO pipeline. In Fig. 2, the two values before and after using PRRO are absolute utility change compared to the baseline as defined in Def. 2.2, where the change shows the relative change of synthetic data utility after implementing the PRRO pipeline.

4.2.1 *Over 20% improvement in all classification accuracy metrics on synthetic replacement using PRRO.*

The improvement in synthetic data utility in the replacement case is exceptionally notable after implementing the PRRO pipeline. The classification accuracy using original data serves as the upper bound of synthetic model utility in replacement scenario, given the fact that synthetic data is synthesized from the original data and hence is a derivative of the original data. Based on this consideration, PRRO pipeline allows a closer resemblance to the upper bound. **On average, it improves all accuracy-related metrics for more than 20%, including a 20.96%, 24.23% and 26.45% improvement in the precision, recall and F1 respectively.** With a 6.92% increase in the AUC score, it confirms that the conditional relationship between the predictor and features becomes more visible after implementing PRRO, as the SL model is able to make predictions with increased confidence. The consistency of outperformance is further supported by a drop of standard deviation for all metrics quantified by a 9.84%, 4.42% and 10.14% drop in the precision, recall and F1 metrics respectively, implying a closer distance to the upper bound. These increases in all accuracy metrics and drops of standard deviation summarize a consistent outperformance of synthetic data utility after PRRO implementation. This uniform improvement points to the value of the Column Conditional ReOrdering module, as the significance lies between setup before and after the pipeline implementation primarily differentiated by the use of the reordering module. The values of the Signal-based Data Pruning module is illustrated in the more material improvement over the imbalanced datasets, with increase in accuracy being over 30%, as the module is specifically designed to process imbalanced datasets. Further benefits of the Signal-based Data Pruning module are discussed in Sec. 4.2.3, where the detailed label distribution of datasets is also explored.

4.2.2 *A 6.13% improvement in F1 score on synthetic appendant using PRRO.*

In terms of the synthetic data utility under synthetic appendant case, an improvement of synthetic data quality is still observed over all evaluation metrics, despite being not as significant as those in the synthetic replacement case. It is observed that the implementation of PRRO still leads to 6.13% improvement, hence showing the value of the Colunn Conditional ReOrdering module. However, unlike the replacement case, the baseline with original data only should serve as a soft lower bound, as synthetic data is added on top of the original data. With a slight 0.07% drop on the F1 score, the addition of synthetic data does not improve overall data quality in fact. Contradictorily, the degradation of classification performance indicates SL model confusion by the addition of synthetic data. Nevertheless, by analyzing further on the imbalanced datasets, we can observe a more concrete synthetic data utility. To begin with, there is a 13.87% improvement of F1 score after PRRO implementation, and more importantly, 1.28% increase compared to the baseline, implying that PRRO can assist synthetic data in improving supervised learning utility in imbalanced scenario. This can be reasoned with the usage of Signal-based Data Pruning module. By adding synthetic minority data (Sec. 3.1) to the original dataset to distort the original class distribution (Table 1), the SL model can weight more heavily on minority classes with an increasing exposure of the minorities. The significant drop in standard deviation of 11.73% further supports the module consistency under different datasets. A notable observation is that the 1.28% improvement over the baseline is contributed by a 3.82% increase in recall and a 1.04% drop in precision, which is precisely the impact of the Signal-based Data Pruning module. With a distorted data distribution weighting the minority classes more heavily, this leads to a higher probability of false positive over false negative, i.e. the precision-recall trade-off. However, it is generally accepted that, under an imbalanced dataset scenario, the benefit of improving recall (identifying more instances of the minority class) typically outweighs the drawbacks of a potential increase in false positives (precision). The benefit is reflected by the net improvement of F1 score. Nevertheless, there are still cases where precision is more important than recall even at highly imbalanced datasets [30], such as amputation analysis, where the cost of false positive immensely outweighs that of false positive. All in all, the exact balance between

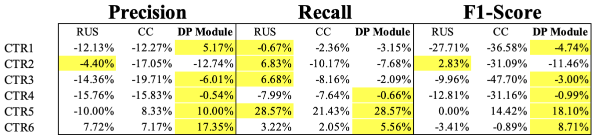


		Datasets	Precision			Recall			F1			AUC		
			No PRRO	⇒ PRRO	Change	No PRRO	⇒ PRRO	Change	No PRRO	⇒ PRRO	Change	No PRRO	⇒ PRRO	Change
Replacement of Original Data	Balanced Datasets	AB	-8.82%	-3.98%	5.30%	-0.29%	-7.84%	-7.57%	-4.99%	-5.84%	-0.90%	-4.93%	-5.32%	-0.40%
		AC	-74.06%	-34.72%	151.67%	-72.22%	-16.67%	200.00%	-73.21%	-27.17%	171.88%	-18.72%	-20.58%	-2.28%
		CG	-87.24%	-5.90%	637.21%	-94.59%	-54.23%	746.67%	-92.38%	-38.27%	710.51%	-45.06%	-17.37%	50.41%
		DI	-11.10%	-1.62%	10.66%	19.32%	1.07%	-15.29%	2.15%	-0.27%	-2.37%	-8.01%	-2.64%	5.83%
		FS	1.15%	-2.86%	-3.96%	-16.69%	-4.42%	14.73%	-8.63%	-3.64%	5.46%	-3.67%	-2.37%	1.35%
		HA	-15.13%	0.19%	18.04%	-42.64%	15.33%	101.06%	-31.00%	6.92%	54.96%	-45.82%	15.49%	113.17%
		HE	-35.26%	-43.60%	-12.89%	20.56%	24.74%	3.47%	-17.71%	-24.57%	-8.34%	-13.72%	-30.64%	-19.60%
		HF	-29.73%	-12.40%	24.66%	9.62%	-18.27%	-25.44%	-12.34%	-15.74%	-3.88%	-12.77%	-14.17%	-1.61%
		LI	-12.38%	-55.96%	-49.74%	51.48%	-46.23%	-64.50%	8.77%	-51.95%	-55.82%	-4.08%	-23.48%	-20.22%
		ST	-9.30%	-7.99%	1.44%	7.50%	-2.72%	-9.50%	-1.62%	-5.43%	-3.88%	-15.44%	-12.99%	2.90%
		TI	-34.43%	-21.41%	19.86%	-55.18%	-1.52%	119.73%	-46.84%	-12.50%	64.60%	-32.53%	-13.26%	28.56%
		WQ	-2.04%	-6.05%	-4.09%	-15.63%	-54.09%	-45.59%	-9.40%	-38.51%	-32.13%	-5.77%	-14.78%	-9.56%
		Average	-26.53%	-16.36%	**13.84%**	-15.73%	-13.74%	2.37%	-23.93%	-18.08%	7.69%	-17.54%	-11.84%	6.92%
		Best	1.15%	0.19%	-0.95%	51.48%	24.74%	-17.65%	8.77%	6.92%	-1.70%	-3.67%	15.49%	**19.89%**
		SD	28.02%	18.59%	-7.37%	42.83%	25.74%	**-11.96%**	31.50%	18.07%	**-10.21%**	15.34%	11.97%	-2.92%
	Imbalanced Datasets	AD	4.31%	9.01%	4.51%	-61.01%	-68.78%	-19.93%	-44.47%	-52.74%	-14.89%	-9.76%	-5.32%	4.93%
		BL	-9.26%	-7.70%	1.72%	2.73%	37.70%	34.04%	-2.64%	14.25%	17.34%	1.57%	-6.76%	-8.20%
		CR	-75.67%	10.80%	355.46%	-94.05%	-27.10%	1125.60%	-91.21%	-14.59%	871.46%	-22.58%	-4.03%	23.97%
		MC	-57.72%	-47.97%	23.04%	36.99%	26.42%	-7.72%	-30.83%	-22.24%	12.41%	-18.42%	-11.32%	8.71%
		CTR1	15.69%	50.18%	29.81%	-20.29%	-17.06%	4.05%	-4.98%	8.00%	13.65%	-9.68%	-6.83%	3.16%
		CTR2	26.12%	-25.00%	-40.53%	-70.00%	88.93%	529.76%	-52.83%	9.48%	132.09%	-17.58%	-2.27%	18.58%
		CTR3	-92.36%	5.27%	1277.85%	-55.77%	-12.03%	98.90%	-86.04%	-4.96%	580.60%	-22.90%	-3.95%	24.59%
		CTR4	-89.95%	-13.81%	757.33%	-56.86%	-9.06%	110.82%	-81.57%	-11.06%	382.58%	-21.65%	-2.32%	24.67%
		CTR5	38.21%	-36.00%	-53.70%	-26.32%	-5.26%	28.57%	2.83%	-26.65%	-28.67%	43.22%	-30.65%	-51.58%
		CTR6	-91.64%	-74.32%	207.35%	-64.71%	-58.82%	16.67%	-86.13%	-68.06%	130.34%	-28.87%	-25.36%	4.93%
		Average	-33.23%	-12.95%	**30.36%**	-40.93%	-4.51%	**61.66%**	-47.79%	-16.86%	**59.23%**	-10.67%	-9.88%	0.88%
		Best	38.21%	50.18%	8.66%	36.99%	88.93%	**37.91%**	2.83%	14.25%	**11.11%**	43.22%	-2.27%	-31.76%
		SD	53.26%	35.06%	**-11.88%**	39.06%	46.30%	5.21%	37.59%	26.86%	-7.80%	20.83%	9.99%	-8.98%
	Overall Average		-29.57%	-14.81%	**20.96%**	-27.19%	-9.54%	**24.23%**	-34.78%	-17.52%	**26.45%**	-14.42%	-10.95%	4.05%
	Overall Best		38.21%	50.18%	8.66%	51.48%	88.93%	**24.73%**	8.77%	14.25%	5.04%	43.22%	15.49%	-19.36%
	Overall SD		40.48%	26.66%	-9.84%	42.18%	35.89%	-4.42%	35.68%	21.93%	**-10.14%**	17.93%	10.90%	-5.96%

		Datasets	Precision			Recall			F1			AUC		
			No PRRO	⇒ PRRO	Change	No PRRO	⇒ PRRO	Change	No PRRO	⇒ PRRO	Change	No PRRO	⇒ PRRO	Change
Appendant to Original Data	Balanced Datasets	AB	-1.78%	-0.04%	1.77%	0.13%	-1.10%	-1.23%	-0.89%	-0.54%	0.35%	-1.87%	-1.33%	0.55%
		AC	-19.54%	-2.92%	20.64%	-24.07%	-12.96%	14.63%	-21.77%	-8.00%	17.60%	-7.20%	0.00%	7.76%
		CG	-1.38%	0.56%	1.97%	-3.78%	-10.27%	-6.74%	-2.59%	-5.12%	-2.60%	-1.72%	-0.89%	0.85%
		DI	-2.01%	-0.72%	1.31%	7.56%	0.38%	-6.68%	2.64%	-0.17%	-2.73%	-0.14%	-0.50%	-0.36%
		FS	0.97%	1.71%	0.73%	-3.56%	-0.33%	3.35%	-1.34%	0.68%	2.05%	-0.59%	-0.48%	0.11%
		HA	-0.50%	1.64%	2.16%	-0.15%	2.58%	2.74%	-0.34%	2.09%	2.43%	-7.92%	13.81%	23.60%
		HE	2.63%	3.86%	1.20%	4.53%	-2.44%	-6.67%	3.50%	0.86%	-2.55%	-0.06%	-7.38%	-7.32%
		HF	-7.24%	-4.18%	3.30%	16.35%	-3.16%	-16.77%	4.46%	-3.62%	-7.74%	-1.31%	-7.85%	-6.63%
		LI	-7.76%	-0.16%	8.24%	17.38%	-2.11%	-16.60%	2.35%	-1.07%	-3.34%	2.98%	-3.93%	-6.71%
		ST	1.41%	0.68%	-0.72%	2.16%	0.09%	-2.02%	1.78%	0.39%	-1.37%	0.74%	0.08%	-0.66%
		TI	-1.01%	-1.21%	-0.21%	-3.66%	2.44%	6.33%	-2.36%	0.60%	3.03%	-3.20%	-1.76%	1.49%
		WQ	-2.13%	2.28%	4.50%	-4.76%	-3.08%	1.76%	-3.47%	-0.50%	3.08%	-1.80%	-0.28%	1.55%
		Average	-3.19%	0.12%	3.43%	0.68%	-2.50%	-3.15%	-1.50%	-1.20%	0.31%	-1.84%	-0.88%	0.99%
		Best	2.63%	3.86%	1.20%	17.38%	2.58%	-12.61%	4.46%	2.09%	-2.27%	2.98%	13.81%	**10.52%**
		SD	6.00%	2.22%	-3.57%	10.82%	4.70%	-5.53%	6.89%	2.92%	-3.71%	3.10%	5.39%	2.22%
	Imbalanced Datasets	AD	-7.39%	2.87%	11.08%	-19.28%	-14.23%	6.25%	-14.03%	-6.87%	8.33%	-3.88%	-4.52%	-0.67%
		BL	6.10%	14.50%	7.92%	11.48%	19.13%	6.86%	9.17%	17.15%	7.31%	0.94%	-2.46%	-3.37%
		CR	-6.06%	0.66%	7.16%	-28.59%	-2.64%	36.33%	-20.42%	-1.25%	24.08%	-2.30%	-1.89%	0.42%
		MC	-21.05%	-8.23%	16.24%	-9.35%	3.25%	13.90%	-14.87%	-2.11%	14.99%	-3.71%	-4.00%	-0.31%
		CTR1	-8.09%	-1.79%	6.85%	1.26%	-1.62%	-2.84%	-3.81%	-1.71%	2.19%	-4.83%	-3.68%	1.21%
		CTR2	-5.81%	2.52%	8.84%	-1.79%	8.57%	10.55%	-3.75%	5.59%	9.71%	-7.56%	-0.35%	7.79%
		CTR3	-3.12%	-1.13%	2.05%	17.72%	7.32%	-8.84%	7.28%	3.32%	-3.69%	-1.23%	0.07%	1.31%
		CTR4	-9.93%	-3.31%	7.35%	3.55%	7.19%	3.52%	-2.40%	2.65%	5.17%	-5.18%	0.41%	5.89%
		CTR5	0.00%	-8.78%	-8.78%	-5.26%	-5.26%	0.00%	-2.14%	-7.43%	-5.41%	-0.50%	-15.58%	-15.15%
		CTR6	-78.41%	-7.69%	327.60%	-23.53%	16.53%	52.38%	-65.53%	3.51%	200.32%	-14.79%	4.90%	23.11%
		Average	-13.38%	-1.04%	**14.24%**	-5.38%	3.82%	9.73%	-11.05%	1.28%	**13.87%**	-4.30%	-2.71%	1.66%
		Best	6.10%	14.50%	7.92%	17.72%	19.13%	1.19%	9.17%	17.15%	7.31%	0.94%	4.90%	3.93%
		SD	23.89%	6.94%	**-13.68%**	15.02%	10.12%	-4.26%	21.30%	7.08%	**-11.73%**	4.44%	5.30%	0.83%
	Overall Average		-7.82%	-0.40%	8.05%	-2.08%	0.38%	2.50%	-5.84%	-0.07%	6.13%	-2.96%	-1.71%	1.29%
	Overall Best		6.10%	14.50%	7.92%	17.72%	19.13%	1.19%	9.17%	17.15%	7.31%	2.98%	13.81%	**10.52%**
	Overall SD		17.04%	4.86%	**-10.41%**	12.95%	8.11%	-4.28%	15.59%	5.25%	-8.95%	3.88%	5.30%	1.37%

Figure 2: The graph shows SL utility comparison between original and synthetic data. Bold numbers indicate that the change after implementing PRRO pipeline that results in a more than 10% improvement for illustrative purpose. Synthetic data utility is evaluated in two aspects: (1) replacement: whether the synthetic data alone can perform as well as the original data, and (2) appendant: whether the addition of synthetic data can further enhance prediction performance of the original data. It is notable that PRRO implementation leads to a significant increase in both the average and best cases of classification metrics but a decrease in the standard deviation, summarizing a consistent and significant improvement in synthetic data utility after implementing PRRO.

precision and recall varies across different contexts and datasets. Moreover, we compare the efficacy of the Data Pruning module against various undersampling methods including Random Undersampling and Cluster Centroids [25, 76], where Fig. 3 shows that synthetic data generated under our Data Pruning module has a consistent outperformance on classification F1-Score, as well as precision.



	Precision			Recall			F1-Score		
	RUS	CC	**DP Module**	RUS	CC	**DP Module**	RUS	CC	**DP Module**
CTR1	-12.13%	-12.27%	5.17%	-0.67%	-2.36%	-3.15%	-27.71%	-36.58%	-4.74%
CTR2	-4.40%	-17.05%	-12.74%	6.83%	-10.17%	-7.68%	2.83%	-31.09%	-11.46%
CTR3	-14.36%	-19.71%	-6.01%	6.68%	-8.16%	-2.09%	-9.96%	-47.70%	-3.00%
CTR4	-15.76%	-15.83%	-0.54%	-7.99%	-7.64%	-0.66%	-12.81%	-31.16%	-0.99%
CTR5	-10.00%	8.33%	10.00%	28.57%	21.43%	28.57%	0.00%	14.42%	18.10%
CTR6	7.72%	7.17%	17.35%	3.22%	2.05%	5.56%	-3.41%	-0.89%	8.71%

Figure 3: Conducting the same experiments with two undersampling techniques, i.e. Random Undersampling ('RUS') and Cluster Centrois ('CC') shows that the Data Pruning Module (Sec. 3.1) can effectively outperform existing undersampling techniques in terms of synthesizing data with high supervised learning utility.

Datasets	Without Pruning		With Pruning	
	Original	Synthetic	Original	Synthetic
CDR1	0.9777 %	0.0889%	4.1448%	0.4306%
CDR2	1.6568%	0.0200%	9.9960%	14.9338%
CDR3	1.2662%	0.0015%	5.0711%	0.7652%
CDR4	2.1325%	0.0018%	9.1426%	6.0823%
CDR5	4.0541%	5.4054%	26.3158%	31.5789%
CDR6	1.6252%	0.1912%	13.0769%	8.4615%
Average	1.9521%	0.9515%	11.2912%	10.3754%

Table 1: Pruning does not only improve the positive rate for subsequent data analytics ease, but also allows the generator to generate minority class at a rate more similar to the original data. Colored synthetic cases had a 0% positive rate in fact, which made subsequent supervised learning utility evaluation impossible. Therefore, one random point was arbitrarily chosen to be positive, contributing to the non-zero positive rate.

4.2.3 43% improvement in supervised learning utility after data pruning.

The Signal-based Data Pruning module specifically improves synthetic data utility of imbalanced datasets. When studying the positive rate of the extremely imbalanced subdatasets from [12], Table 1 shows discount in minority distribution in synthetic data compared to original data. The module significantly reduces the rate of discount after pruning. This does not only quantify the improvement of class distribution similarity, but also contributes to synthetic data utility improvement in Fig. 2. Precisely, the positive rate is observed to be extremely imbalanced where none of the positive rate is above 5% and the average is as low as 1.95%. **Before data pruning**, the synthetic positive rate is **discounted by** 51% from 1.95% to 0.95%, exaggerated by the generator. **After pruning**, with the positive rate improving to 11.29%, **a significantly smaller 8% discount rate** from 11.29% to 10.38% can be observed. This accounts for **a 43% improvement** in similarity of class distribution between original and synthetic data. This drop in discount rate highlights more similar class distribution between original and synthetic data, hence underscoring the value of signal-based data pruning method in handling highly imbalanced datasets.

5 Summary and Future Work

In this work, we introduce the Pruning and Reordering PRRO pipeline, a novel data-centric framework designed to enhance the utility of synthetic data generated by tabular data generators. Our approach systematically addresses key challenges associated with generators that exaggerate imbalance class distribution and overlook necessary conditional data relationships of supervised learning models, tackling the challenges with the Signal-based Data Pruning and Column Conditional ReOrdering modules respectively. Our experimental results across 22 diverse datasets show that synthetic data utility improvement after implementing the PRRO pipeline. Precisely, the Column Conditional ReOrdering module allows a generalized improvement of synthetic data utility, after addressing the generator's hurdle in overlooking predictor-features conditional relationship. Furthermore, the synthetic data utility is further enhanced in the case of imbalanced datasets thanks to the Signal-based Data Pruning module, which addresses the imbalance dataset exaggeration within generators. The PRRO pipeline allows generator to synthesize more original-data alike synthetic data to accomplish an improved efficacy in the synthetic replacement task. While synthetic data utility improvement is more material in synthetic replacement scenario, the PRRO also contributes to a betterment of synthetic appendant to minority class, which allows SL model trained under the newly combined dataset to capture patterns of minority class more easily. All in all, the PRRO pipeline bridges the critical gaps between state-of-the-art generators and SL models, streamlining the 'generate and predict' process.

Looking ahead, we see possible utility improvement by further aligning the data structure between tabular data and LLMs. Other than the textual encoding technique that converts each row observation into a sentence [3, 60], there are also other works seeking to effectively encode tables for LLM fine-tuning and training [17, 33, 62]. By developing a model structure that can effectively study both column-wise and row-wise relationships across a table, the application of LLM can be immensely extended in the area of tabular data. Aside from developing a tabular encoder that fully complies to tabular data, we can also explore the betterment of tabular modeling with existing LLMs, by leveraging the corresponding position bias to allocate corresponding weights to different column relationships. There have been works exploring the mitigation and alignment of position bias in LLM [34, 72], motivating an further step in handling LLM position bias - utilizing the bias to improve model performance.

Specifically, we propose an extension of the Column Conditional ReOrdering algorithm to adapt to potential multicollinearity problem in real-life datasets. Given the column-by-column conditional relationship modeling of the tabular generators, we hypothesize leveraging this modeling structure by arranging the features ordered by importance to the predictor, so that columns of higher importance can be further emphasized by the column. Algo. 3 proposes reordering the features in terms of the importance, e.g. quantified by permutation importance score from [4]. Next, given the column-by-column conditional relationship adopted by generators, we hypothesize that columns that are modeled in the beginning have less probability constraints, compared to the generation of later columns which have to follow the conditional density of previous features, thereby leading to a higher degree of randomness. Therefore, by placing the more important features at the end, this guides the generator in modeling and synthesizing important features more carefully (with more constraints), preserving data relationship of higher importance and hence enhancing synthetic data utility. With future works hypothesizing improvement in generator models and training data quality, we are devoted to continue advancement in the area of trustworthy data synthesis to promote safer and more accessible data analytics.

References

[1] Anonymous. 2024. Eliminating Position Bias of Language Models: A Mechanistic Approach. In *Submitted to The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=fvkElsJOsN under review.

[2] Tom Archer. 2023. Tech Translated: Synthetic data. https://www.pwc.com/gx/en/issues/technology/synthetic-data.html

[3] Vadim Borisov, Kathrin Sessler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2023. Language Models are Realistic Tabular Data Generators. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=cEygmQNOeI

[4] Leo Breiman. 2001. Random Forest. doi:10.1023/A:1010933404324

[5] Broadstreet. 2022. What is a Successful Click-Through Rate? https://broadstreetads.com/successful-click-through-rate/

[6] Hsin-Yu Chang, Pei-Yu Chen, Tun-Hsiang Chou, Chang-Sheng Kao, Hsuan-Yun Yu, Yen-Ting Lin, and Yun-Nung Chen. 2024. A Survey of Data Synthesis Approaches. arXiv:2407.03672 [cs.LG] https://arxiv.org/abs/2407.03672

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16 (June 2002), 321–357. doi:10.1613/jair.953

[8] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, 785–794. doi:10.1145/2939672.2939785

[9] Xinyun Chen, Ryan A. Chi, Xuezhi Wang, and Denny Zhou. 2024. Premise Order Matters in Reasoning with Large Language Models. arXiv:2402.08939 [cs.AI] https://arxiv.org/abs/2402.08939

[10] Waseem Ahmad Chishti and Shahid Mahmood Awan. 2019. Deep Neural Network a Step by Step Approach to Classify Credit Card Default Customer. In *2019 International Conference on Innovative Computing (ICIC)*. 1–8. doi:10.1109/ICIC48496.2019.8966723

[11] Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum Associates, Hillsdale, NJ. ISBN: 0-8058-0283-5.

[12] DeepInvolution. 2022. CTR Prediction - 2022 DIGIX Global AI Challenge. Kaggle. DOI: https://www.kaggle.com/datasets/xiaojiu1414/digix-global-ai-challenge.

[13] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2024. DeepSeek-V3 Technical Report. arXiv:2412.19437 [cs.CL] https://arxiv.org/abs/2412.19437

[14] Kailaash Devi. 2023. Understanding Hold-Out Methods for Training Machine Learning Models. https://www.comet.com/site/blog/understanding-hold-out-methods-for-training-machine-learning-models/

[15] Richard O. Duda, Peter E. Hart, and David G. Stork. 2000. Pattern Classification. (2000). Chapter on Naive Bayes Classifier, including Gaussian Naive Bayes.

[16] Yotam Elor and Hadar Averbuch-Elor. 2022. To SMOTE, or not to SMOTE? arXiv:2201.08528 [cs.LG] https://arxiv.org/abs/2201.08528

[17] Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. Large Language Models(LLMs) on Tabular Data: Prediction, Generation, and Understanding – A Survey. arXiv:2402.17944 [cs.CL] https://arxiv.org/abs/2402.17944

[18] Candace Flatt and Ronald Jacobs. 2019. Principle Assumptions of Regression Analysis: Testing, Techniques, and Statistical Reporting of Imperfect Data Sets. *Advances in Developing Human Resources* 21 (11 2019), 484–502. doi:10.1177/1523422319869915

[19] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189–1232.

[20] Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and Fairness in Large Language Models: A Survey. arXiv:2309.00770 [cs.CL] https://arxiv.org/abs/2309.00770

[21] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL] https://arxiv.org/abs/2312.10997

[22] Seda Guney, Irem Kilinc, Alaa Hameed, and Akhtar Jamil. 2022. *Abalone Age Prediction Using Machine Learning*. 329–338. doi:10.1007/978-3-031-04112-9_25

[23] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. arXiv:1703.04247 [cs.IR] https://arxiv.org/abs/1703.04247

[24] Román Salmerón Gómez, Catalina García García, and Ainara Rodríguez Sánchez. 2024. Enlarging of the sample to address multicollinearity. arXiv:2407.01172 [stat.AP] https://arxiv.org/abs/2407.01172

[25] Tawfiq Hasanin and Taghi Khoshgoftaar. 2018. The Effects of Random Undersampling with Simulated Class Imbalance for Big Data. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. 70–79. doi:10.1109/IRI.2018.00018

[26] Ahmad B. Hassanat, Ahmad S. Tarawneh, Ghada A. Altarawneh, and Abdullah Almuhaimeed. 2022. Stop Oversampling for Class Imbalance Learning: A Critical Review. arXiv:2202.03579 [cs.LG] https://arxiv.org/abs/2202.03579

[27] iab. Tech Lab. 2024. ADMAP: Attribution Data Matching Protocol: A Data Clean Room Interoperability Protocol for Attribution Measurement. iab. Tech Lab. DOI: https://iabtechlab.com/wp-content/uploads/2024/10/ADMAP-Version-1.0-for-PUBLIC-COMMENT.pdf.

[28] Jing Yu Koh, Daniel Fried, and Ruslan Salakhutdinov. 2023. Generating Images with Multimodal Language Models. arXiv:2305.17216 [cs.CL] https://arxiv.org/abs/2305.17216

[29] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2022. TabDDPM: Modelling Tabular Data with Diffusion Models. https://arxiv.org/abs/2209.15421

[30] Rohit Kundu. 2022. Precision vs. Recall: Differences, Use Cases & Evaluation. https://www.v7labs.com/blog/precision-vs-recall-guide

[31] Tung Sum Thomas Kwok, Chi hua Wang, and Guang Cheng. 2024. DEREC-SIMPRO: unlock Language Model benefits to advance Synthesis in Data Clean Room. arXiv:2411.00879 [cs.DB] https://arxiv.org/abs/2411.00879

[32] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[33] Cheng-Te Li, Yu-Che Tsai, Chih-Yao Chen, and Jay Chiehen Liao. 2024. Graph Neural Networks for Tabular Data Learning: A Survey with Taxonomy and Directions. arXiv:2401.02143 [cs.LG] https://arxiv.org/abs/2401.02143

[34] Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024. Split and Merge: Aligning Position Biases in LLM-based Evaluators. arXiv:2310.01432 [cs.CL] https://arxiv.org/abs/2310.01432

[35] A. McCallum and K. Nigam. 1998. Improving text classification by discovering word dependencies. *Proceedings of the 15th International Conference on Machine Learning (ICML 1998)* (1998), 359–367.

[36] Meta. 2024. Introducing Llama 3.2. https://www.llama.com/

[37] Robert L. Miller. 2005. *Applying Statistical Methods in the Social Sciences* (2nd ed.). Prentice Hall, Upper Saddle River, NJ. ISBN: 0-13-108703-4.

[38] Alhassan Mumuni and Fuseini Mumuni. 2022. Data augmentation: A comprehensive survey of modern approaches. *Array* 16 (11 2022), 100258. doi:10.1016/j.array.2022.100258

[39] Dang Nguyen, Sunil Gupta, Kien Do, Thin Nguyen, and Svetha Venkatesh. 2024. Generating Realistic Tabular Data with Large Language Models. arXiv:2410.21717 [cs.LG] https://arxiv.org/abs/2410.21717

[40] Geoff Norman. 2010. Likert scales, levels of measurement and the 'laws' of statistics. *Advances in Health Sciences Education* 15, 5 (2010), 625–632. doi:10.1007/s10459-010-9222-y

[41] Curtis G. Northcutt, Lu Jiang, and Isaac L. Chuang. 2022. Confident Learning: Estimating Uncertainty in Dataset Labels. arXiv:1911.00068 [stat.ML] https://arxiv.org/abs/1911.00068

[42] Shyam Nuggehalli, Jifan Zhang, Lalit Jain, and Robert Nowak. 2024. DIRECT: Deep Active Learning under Imbalance and Label Noise. arXiv:2312.09196 [cs.LG] https://arxiv.org/abs/2312.09196

[43] Jane Ostler and Ashok Kalidas. 2024. Synthetic Data: The Real Deal? The opportunities and challenges of synthetic data for market research. https://www.kantar.com/inspiration/ai/synthetic-data-the-real-deal

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG] https://arxiv.org/abs/1912.01703

[45] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The Synthetic Data Vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 399–410. doi:10.1109/DSAA.2016.49

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[47] Dehua Peng, Zhipeng Gui, and Huayi Wu. 2024. Interpreting the Curse of Dimensionality from Distance Concentration and Manifold Effect. arXiv:2401.00422 [cs.LG] https://arxiv.org/abs/2401.00422

[48] Joanne Peng, Kuk Lee, and Gary Ingersoll. 2002. An Introduction to Logistic Regression Analysis and Reporting. *Journal of Educational Research - J EDUC RES* 96 (09 2002), 3–14. doi:10.1080/00220670209598786

[49] Vasileios C. Pezoulas, Dimitrios I. Zaridis, Eugenia Mylona, Christos Androutsos, Kosmas Apostolidis, Nikolaos S. Tachos, and Dimitrios I. Fotiadis. 2024. Synthetic data generation methods in healthcare: A review on open-source tools and methods. doi:10.1016/j.csbj.2024.07.005

[50] Michael Platzer and Thomas Reutterer. 2021. Holdout-Based Empirical Assessment of Mixed-Type Synthetic Data. *Frontiers in Big Data* 4 (06 2021), 679939. doi:10.3389/fdata.2021.679939

[51] Zahra Pourbahman, Saeedeh Momtazi, and Alireza Bagheri. 2023. Deep neural ranking model using distributed smoothing. doi:10.1016/j.eswa.2023.119913

[52] Zhaozhi Qian, Thomas Callender, Bogdan Cebere, Sam M. Janes, Neal Navani, and Mihaela van der Schaar. 2024. Synthetic data for privacy-preserving clinical risk prediction. doi:10.1038/s41598-024-72894-y

[53] J. R. Quinlan. 1986. Induction of decision trees. In *Machine Learning*, Vol. 1. Springer, 81–106.

[54] Data Reply. 2023. How synthetic data is revolutionising Industries. https://www.reply.com/data-reply/en/how-synthetic-data-is-revolutionising-industries

[55] Patrick Schober, Christa Boer, and Lothar A Schwarte. 2018. Correlation Coefficients: Appropriate Use and Interpretation. doi:10.1213/ANE.0000000000002864

[56] Sandeep Singh Sengar, Affan Bin Hasan, Sanjay Kumar, and Fiona Carroll. 2024. Generative Artificial Intelligence: A Systematic Review and Applications. arXiv:2405.11029 [cs.LG] https://arxiv.org/abs/2405.11029

[57] Jonathon Shlens. 2014. A Tutorial on Principal Component Analysis. arXiv:1404.1100 [cs.LG] https://arxiv.org/abs/1404.1100

[58] Noora Shrestha. 2020. Detecting Multicollinearity in Regression Analysis. *American Journal of Applied Mathematics and Statistics* 8 (06 2020), 39–42. doi:10.12691/ajams-8-2-1

[59] Aakriti Singh, Shipra Saraswat, and Neetu Faujdar. 2017. Analyzing Titanic disaster using machine learning algorithms. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*. 406–411. doi:10.1109/CCAA.2017.8229835

[60] Aivin V. Solatorio and Olivier Dupriez. 2023. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers. arXiv:2302.02041 [cs.LG] https://arxiv.org/abs/2302.02041

[61] Samuel Stocksieker, Denys Pommeret, and Arthur Charpentier. 2023. Generalized Oversampling for Learning from Imbalanced datasets and Associated Theory. arXiv:2308.02966 [stat.ML] https://arxiv.org/abs/2308.02966

[62] Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, Haoxuan Lan, Jiaming Tian, Jing Yuan, Junbo Zhao, Junlin Zhou, Kaizhe Shou, Liangyu Zha, Lin Long, Liyao Li, Pengzuo Wu, Qi Zhang, Qingyi Huang, Saisai Yang, Tao Zhang, Wentao Ye, Wufang Zhu, Xiaomeng Hu, Xijun Gu, Xinjie Sun, Xiang Li, Yuhang Yang, and Zhiqing Xiao. 2024. TableGPT2: A Large Multimodal Model with Tabular Data Integration. arXiv:2411.02059 [cs.LG] https://arxiv.org/abs/2411.02059

[63] Yasiru Tishan. 2023. Understanding the Difference Between Supervised and Unsupervised Learning Techniques. doi:10.13140/RG.2.2.36176.48641

[64] Weixi Tong and Tianyi Zhang. 2024. CodeJudge: Evaluating Code Generation with Large Language Models. arXiv:2410.02184 [cs.LG] https://arxiv.org/abs/2410.02184

[65] Gil Trotino. 2024. Synthetic Data Use Cases for Every Company. https://www.k2view.com/blog/synthetic-data-use-cases/#Understanding-Synthetic-Data

[66] Artem Vysogorets, Kartik Ahuja, and Julia Kempe. 2024. DRoP: Distributionally Robust Pruning. arXiv:2404.05579 [cs.LG] https://arxiv.org/abs/2404.05579

[67] X. Wei and Y. Zhou. 2005. Complement naive bayes: A new approach to classification of imbalanced data. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2005)*. Springer, 370–380.

[68] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling Tabular data using Conditional GAN. In *Advances in Neural Information Processing Systems*.

[69] Shirong Xu, Will Wei Sun, and Guang Cheng. 2024. Utility Theory of Synthetic Data Generation. arXiv:2305.10015 [stat.ML] https://arxiv.org/abs/2305.10015

[70] Yilin Ye, Jianing Hao, Yihan Hou, Zhan Wang, Shishi Xiao, Yuyu Luo, and Wei Zeng. 2024. Generative AI for Visualization: State of the Art and Future Directions. arXiv:2404.18144 [cs.LG] https://arxiv.org/abs/2404.18144

[71] Moon Ye-Bin, Nam Hyeon-Woo, Wonseok Choi, Nayeong Kim, Suha Kwak, and Tae-Hyun Oh. 2024. SYNAuG: Exploiting Synthetic Data for Data Imbalance Problems. arXiv:2308.00994 [cs.CV] https://arxiv.org/abs/2308.00994

[72] Yijiong Yu, Huiqiang Jiang, Xufang Luo, Qianhui Wu, Chin-Yew Lin, Dongsheng Li, Yuqing Yang, Yongfeng Huang, and Lili Qiu. 2024. Mitigate Position Bias in Large Language Models via Scaling a Single Dimension. arXiv:2406.02536 [cs.CL] https://arxiv.org/abs/2406.02536

[73] Jianhao Yuan, Jie Zhang, Shuyang Sun, Philip Torr, and Bo Zhao. 2024. Real-Fake: Effective Training Data Synthesis Through Distribution Matching. arXiv:2310.10402 [cs.LG] https://arxiv.org/abs/2310.10402

[74] Tongtong Yuan, Weihong Deng, Jian Tang, Yinan Tang, and Binghui Chen. 2019. Signal-to-Noise Ratio: A Robust Distance Metric for Deep Metric Learning. arXiv:1904.02616 [cs.CV] https://arxiv.org/abs/1904.02616

[75] Bo Zhang, Ruotong Yu, Haipei Sun, Yanying Li, Jun Xu, and Hui Wang. 2020. Privacy for All: Demystify Vulnerability Disparity of Differential Privacy against Membership Inference Attack. arXiv:2001.08855 [cs.CR] https://arxiv.org/abs/2001.08855

[76] Yan-Ping Zhang, Li-Na Zhang, and Yong-Cheng Wang. 2010. Cluster-based Majority Under-Sampling Approaches for Class Imbalance Learning. *Proceedings - 2010 2nd IEEE International Conference on Information and Financial Engineering, ICIFE 2010* (09 2010). doi:10.1109/ICIFE.2010.5609385

A Appendix

A.1 Textual Encoding Transformation for LLM-based Tabular Synthesis

Textual encoding transformation $T(\cdot)$ is a method proposed in [3] that transform tabular observations into natural language for LLM processing. Given a dataset with n observations and k features ($\mathcal{D} = \{x_{i1}, x_{i2}, ..., x_{ik}\}_{i=1}^{n}$, each observation is converted into a sentence as follows: 'Column 1: x_{i1}, Column 2: x_{i2}, ..., Column k: x_{ik}', using comma (,) and colon (:) for separation between columns and within column name and column value. The sentences are then used to fine-tune the LLM to generate texts that mimic the behavior of original texts. These semantic-preserving synthetic sentences are inversely transformed back to the tabular form [3].

A.2 Detailed summary for the datasets used in Sec. 4 Experiment

Abbr.	Name	# Train (Holdout)	# Test	# Num	# Cat	Positive Rate
AB	Abalone	1,670	836	7	1	49.82%
AC	Acute Inflammation	48	24	1	7	39.58%
AD	Adult Income	14,652	7,327	5	8	23.93%
BL	Blood Donation	229	150	3	0	23.75%
CG	Congress	174	87	0	16	38.51%
CR	Credit	9,600	4,800	14	9	22.18%
DI	Diabetes	28,276	14,139	3	18	50.00%
FS	Flight Satisfaction	41,561	20,781	4	18	43.33%
HA	Haberman	122	62	3	0	73.77%
HE	Heart Disease	104	53	5	5	37.5%
HF	Heart Failure	119	60	7	5	31.93%
LI	Liver Disease	233	177	9	1	28.76%
MC	Marketing Campaign	896	448	17	10	14.96%
ST	Student Performance	400	200	2	5	51.5%
TI	Titanic	356	179	2	7	38.48%
WQ	Wine Quality	457	229	11	0	54.27%
CTR1	CTR Group 1	31,503	15,752	4	19	0.98%
CTR2	CTR Group 2	15,029	7,515	4	19	1.66%
CTR3	CTR Group 3	65,946	32,974	4	19	1.27%
CTR4	CTR Group 4	55,756	27,878	4	19	2.13%
CTR5	CTR Group 5	148	75	4	19	4%
CTR6	CTR Group 6	1,041	524	4	19	1.63%

Table 2: Summary of the datasets used for the experiment to test the PRRO pipeline

A.3 Extension of Column Conditional ReOrdering algorithm for multicollinearity adaptation

Algorithm 3 Finding optimal order for generator

```
Input: A training and a validation dataset with $p$ features: $D_{T/V} = \{(\mathbf{x}_{T/V,i}, y_{T/V,i})\}_{i=1}^{k_{T/V}}$, where $\mathbf{x}_{T/V,i} = (x_{T/V,i}^{(1)}, \ldots, x_{T/V,i}^{(p)})$.
Output: Optimal feature order that maximizes utility of generator $\mathcal{G}_A$.
Initialize generator $\mathcal{G}_A$ predictive model $h$, $err \leftarrow \infty$;
$X \leftarrow \mathbf{x}_T$;
$y \leftarrow y_T$;
Train $h$ with $X$ and $y$;
Rank $X$ according to permutation importance score to get the ranking $r$;
Return $r$;
$X \leftarrow X[r]$ {Reorder $X$ according to the ranking};
Train $\mathcal{G}_A$ by concatenating $X$ and $y$;
Synthesize $\tilde{X}$, $\tilde{y}$;
Measure utility between $\{\tilde{X}, \tilde{y}\}$ and $\{X, y\}$ using validation dataset $D_V$;
```

Received 1 February 2025