Benchmarking Synthetic Tabular Data: A Multi-Dimensional Evaluation Framework

Andrey Sidorenko, Michael Platzer, Mario Scriminaci, and Paul Tiwald
{`andrey.sidorenko, michael.platzer, mario.scriminaci, paul.tiwald`}`@mostly.ai`

MOSTLY AI

Abstract

Evaluating the quality of synthetic data remains a key challenge for ensuring privacy and utility in data-driven research. In this work, we present an evaluation framework that quantifies how well synthetic data replicates original distributional properties while ensuring privacy. The proposed approach employs a holdout-based benchmarking strategy that facilitates quantitative assessment through low- and high-dimensional distribution comparisons, embedding-based similarity measures, and nearest-neighbor distance metrics. The framework supports various data types and structures, including sequential and contextual information, and enables interpretable quality diagnostics through a set of standardized metrics. These contributions aim to support reproducibility and methodological consistency in benchmarking of synthetic data generation techniques. The code of the framework is available at https://github.com/mostly-ai/mostlyai-qa.

1 Introduction

Generative Artificial Intelligence (AI) is rapidly transforming data-centric research fields, transcending its initial prominence in unstructured data domains, such as natural language processing and image synthesis, to structured and semi-structured data contexts prevalent within organizational data assets. Synthetic data generation specifically addresses critical challenges, including privacy-preserving data sharing, representation enhancement of underrepresented subpopulations, simulation of rare but consequential scenarios, and imputation of missing data [1, 2, 3]. However, the practical utility and acceptance of generative synthetic data critically depend on a rigorous evaluation of its fidelity (accuracy of representation) and novelty (degree of originality).

Despite the existence of numerous evaluation frameworks for synthetic data [4, 5, 6, 7, 8, 9, 10, 11, 12, 13], comprehensive and accessible tools addressing both fidelity and novelty simultaneously remain scarce. See Table 1 for a high-level tool comparison. In particular, existing tools often emphasize one evaluation dimension at the expense of the other, yielding either high fidelity through replication or high

1

novelty through randomness, but rarely balancing the two dimensions effectively. For instance, merely copying original samples yields high accuracy without being novel, while generating entirely random samples scores high on novelty without being accurate. The true challenge of privacy-safe synthetic data lies in the generation of data that is both accurate *and* novel. Thus, any quality assurance for synthetic data must measure *both* of these dimensions. To fill this methodological void, we introduce `mostlyai-qa`,

Python package	License	HTML	Plots	Metrics	Novelty	Data
`mostlyai-qa` (2024)[a]	Apache	✓	✓	✓	✓	flexible
`ydata-profiling` (2023)[b]	MIT	✓	✓	–	–	flexible
`sdmetrics` (2023)[c]	MIT	–	✓	✓	✓	flexible
`synthcity` (2023)[d]	Apache	–	✓	✓	✓	flexible
`sdnist` (2023)[e]	Permissive	✓	✓	✓	∼	fixed

[a] https://github.com/mostly-ai/mostlyai-qa
[b] https://github.com/ydataai/ydata-profiling
[c] https://github.com/sdv-dev/SDMetrics
[d] https://github.com/vanderschaarlab/synthcity
[e] https://github.com/usnistgov/SDNist

Table 1: Comparison across open-source Python libraries for assessing synthetic data.

an open-source Python framework explicitly designed to comprehensively evaluate the quality of synthetic data. The framework uniquely integrates accuracy, similarity, and novelty metrics within a unified evaluation framework. It effectively handles diverse data types, including numerical, categorical, datetime, and textual, as well as data with missing values and variable row counts per sample, accommodating multi-sequence, multivariate time-series data[1]. The quality of synthetic data is also evaluated by taking into account any contextual data.

The primary contributions of this paper include:

- A novel evaluation framework that simultaneously assesses fidelity and novelty of synthetic datasets.

- Support for comprehensive, automated assessment and visualization of mixed-type data quality.

- Open-source availability under the Apache License v2, promoting broad adoption and collaborative enhancement within the research community.

2   A framework for evaluation of synthetic data

The evaluation of synthetic data requires careful consideration of two primary dimensions: fidelity and novelty. Fidelity describes the degree to which synthetic samples represent the statistical properties of original data, while novelty ensures that generated samples are distinct enough to preserve privacy and avoid direct replication.

---

[1]Multi-sequence time-series data is the predominant structure for behavioral data, where multiple events for multiple individuals are recorded.

The framework combines these concepts by employing an empirical holdout-based assessment for synthetic mixed-type data, introduced in [6]. In that framework, the quality of synthetic data is benchmarked against holdout data samples that were not used in privacy-preserving training, expecting models to produce novel samples that reflect the underlying data distribution without direct replication. Accordingly, synthetic samples should be as close to training samples as holdout samples but not closer. This approach, akin to the use of holdout samples for supervised learning, enables the evaluation of a generative model's ability to generalize underlying patterns rather than merely memorizing specific training samples. The framework is built upon that

Figure 1: An example of metrics summary generated by the framework.

framework and structured around three interrelated categories of metrics - Accuracy, Similarity, and Distances - each comprising specific submetrics that collectively address the dual objectives of fidelity and novelty. Accuracy quantifies lower-dimensional, and similarity higher-dimensional fidelity, whereas the set of distance metrics helps to gauge the novelty of samples (see Fig. 1).

2.1   Accuracy

Accuracy metrics assess how closely synthetic data replicate the low-order marginal, joint distributions, and consistency along the time dimension (sequential data coherence) of the original dataset, with a score of 100% representing an exact match. The overall accuracy score is computed as 100% minus the top-k total variation distance (with k=10) aggregated across three components:

- **Univariate accuracy**: Measures fidelity of discretized univariate distributions across all attributes.

- **Bivariate accuracy**: Captures alignment between pairs of attributes via discretized bivariate frequency tables.

- **Coherence**: Evaluates attribute consistency across sequential records, applicable only to sequential data.

To evaluate low-order marginals, univariate distributions (Fig. 2) and pairwise correlations between columns (bivariate distributions; Fig. 3) are compared. For datasets containing mixed data types, numerical and date-time columns are transformed by discretizing their values into deciles based on the original training data, creating ten equally populated groups per column. For categorical columns, only the ten most

frequent categories are retained, while the less common ones are excluded. This method enables a consistent comparison across different data types, emphasizing the most informative features of the data. For each feature, we derive two vectors of length
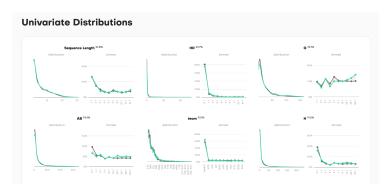

Figure 2: An example of univariate distributions and their accuracies generated by the framework.

10, one from the original training data and one from the synthetic data. In the case of numerical and date-time columns, these vectors capture the frequency of values within the decile-based groups defined by the original data. For categorical columns, the vectors represent the re-normalized frequency distribution of the top ten most frequent categories. These feature-specific vectors are denoted as $\mathbf{X}_{\text{trn}}^{(m)}$ and $\mathbf{X}_{\text{syn}}^{(m)}$, corresponding to the training and synthetic data, respectively. m is the feature index, running from 1 to d.

The **univariate accuracy** of column m is then defined as

$$acc_{\text{univariate}}^{(m)} = \frac{1}{2} \left(1 - \|\mathbf{X}_{\text{trn}}^{(m)} - \mathbf{X}_{\text{syn}}^{(m)}\|_1 \right) \tag{1}$$

and the overall univariate accuracy, as reported in the results section, is defined by

$$acc_{\text{univariate}} = \frac{1}{D} \sum_{m}^{D} acc_{\text{univariate}^{(m)}} \ , \tag{2}$$

where D is the number of columns.

For bivariate metrics, the relationships between pairs of columns are assessed using normalized contingency tables. These tables represent the joint distribution of two features, m and n, enabling the evaluation of pairwise dependencies.

The contingency table between columns m and n is denoted as $\mathbf{C}_{\text{trn}}^{(m,n)}$ for the training data and $\mathbf{C}_{\text{syn}}^{(m,n)}$ for the synthetic data. Each table has a maximum dimension of 10×10, corresponding to the (discretized) values or the top ten categories of the two features. For columns with fewer than ten categories (categorical columns with cardinality <10), the dimensions of the table are reduced accordingly.

Each cell in the table represents the normalized frequency with which a specific combination of categories or discretized values from columns m and n appears in the
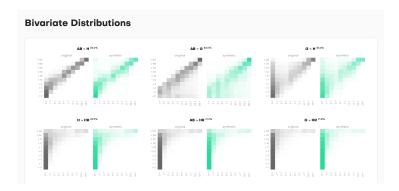
Figure 3: Bivariate distributions and their accuracies generated by the framework.

data. This normalization ensures meaningful comparisons across different features and datasets, independent of their original scale or size.

The **bivariate accuracy** of the column pair m, n is then defined as

$$acc_{\text{bivariate}}^{(m,n)} = \frac{1}{2} \left(1 - \|\mathbf{C}_{\text{trn}}^{(m,n)} - \mathbf{C}_{\text{syn}}^{(m,n)}\|_{1,\text{entrywise}} \right)$$

$$= \frac{1}{2} \left(1 - \sum_i \sum_j \left| \mathbf{C}_{\text{trn}}^{(m,n)} - \mathbf{C}_{\text{syn}}^{(m,n)} \right|_{i,j} \right) \tag{3}$$

and the overall bivariate accuracy, as reported in the results section, is given by

$$acc_{\text{bivariate}} \frac{2}{D(D-1)} \sum_{1 \le m < n \le D} acc_{\text{bivariate}}^{(m,n)} , \tag{4}$$

the average of the strictly upper triangle of $acc_{\text{bivariate}}^{(m,n)}$.

Note that due to sampling noise, both $acc_{\text{univariate}}$ and $acc_{\text{bivariate}}$ cannot reach 1 in practice. The software package reports the theoretical maximum alongside both metrics.

There is no difference in calculating the univariate and bivariate accuracies between flat and sequential data. In both cases, the vectors $\mathbf{X}^{(m)}$ and contingency tables $\mathbf{C}^{(m,n)}$ are based on all entries in the columns, irrespective of which data subject they belong to. For sequential data, the framework evaluates the consistency (coherence) of relationships between successive time steps or sequence elements (Fig. 4). This allows the assessment of whether the original sample autocorrelations within sequences are faithfully reproduced in the synthetic data. The process is as follows:

- For each data subject, we randomly sample two successive sequence elements (time steps) from their sequential data.

- These pairs of successive time steps are transformed into a wide-format dataset. To illustrate, consider a sequential dataset of N subjects and original columns A, B, C, represented as $K > N$ rows. After processing, the resulting dataset
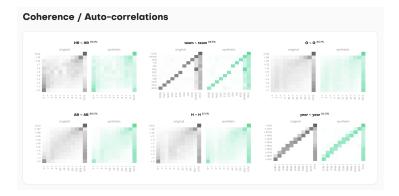
Figure 4: An example of coherence distributions and their accuracies generated by the framework.

has six columns: A, A', B, B', C, C'. The unprimed columns correspond to the first sampled sequence element, the primed columns correspond to the successive sequence element. The number of rows in this wide-format dataset is equal to N, irrespective of the sequence lengths in the original dataset.

Using this wide-format dataset, we construct contingency tables $\mathbf{C}^{(m,m')}$ for each pair of corresponding unprimed and primed columns (m, m'). These tables are normalized and used to calculate the **coherence metric** for column m as:

$$acc_{\text{coherence}}^{(m,m')} = \frac{1}{2} \left(1 - \|\mathbf{C}_{\text{trn}}^{(m,m')} - \mathbf{C}_{\text{syn}}^{(m,m')}\|_{1,\text{entrywise}} \right) \tag{5}$$

and the overall coherence metric, as reported in the results section

$$acc_{\text{coherence}} = \frac{1}{D} \sum_m^D acc_{\text{coherence}}^{(m,m')} . \tag{6}$$

We summarize the **overall accuracy** of a data set as

$$\frac{1}{2} \left(acc_{\text{univariate}} + acc_{\text{bivariate}} \right) \tag{7}$$

and

$$\frac{1}{3} \left(acc_{\text{univariate}} + acc_{\text{bivariate}} + acc_{\text{coherence}} \right) \tag{8}$$

for flat and sequential data, respectively.

This approach offers consistency across attribute types. Additionally, the overall accuracy metric is decomposable into 1-way and 2-way frequency tables, which are visualized as density and heat-map plots, respectively, making it easily interpretable also for non-statisticians. The greater the discrepancies between the plotted distributions, the lower the accuracy score. To achieve a high overall accuracy, each contributing distribution must align closely with the original. However, due to sampling noise with

finite samples, some discrepancies are inevitable. By calculating the expected accuracy for a theoretical holdout dataset based on the original distributions and sample size, we provide a reference benchmark. Rather than aiming for 100% accuracy, the goal is for synthetic samples to match this benchmark closely, indicating similarity to the training samples akin to holdout samples.

When contextual data is present, the framework will report the accuracy of bivariate distributions between contextual and target attributes, enabling assessment of whether these relationships are well-preserved in the synthetic data.

2.2 Centroid Similarity

Complementing accuracy, we report another set of metrics that assess the similarity of distributions. Rather than analyzing the easy-to-interpret lower-dimensional marginals, the focus shifts to the high-dimensional full joint distributions. Direct analysis of high-dimensional distributions is not feasible due to the curse of dimensionality, so we use an alternative approach. Every tabular sample is first converted into a string of values (e.g., `value_col1;value_col2;...;value_colD`), that is then mapped into an informative embedding space using a pre-trained language model. For sequential data, the string is constructed by concatenating the values of all columns across time steps. For instance, values from time step two are appended to the string containing values from time step one, and so on. For long sequences, the resulting input string is truncated to fit within the language model's context window.

While the choice of language model is flexible, we specifically opted for `all-MiniLML6v2`[2] as it is a lightweight, compute-efficient universal model. It transforms each string of values into a 384-dimensional embedding space. Then centroids for each group of embeddings are calculated as

$$\mathbf{c}_{\text{syn}} = \frac{1}{n_{\text{syn}}} \sum_{i=1}^{n_{\text{syn}}} \mathbf{x}_{\text{syn},i}, \quad \mathbf{c}_{\text{trn}} = \frac{1}{n_{\text{trn}}} \sum_{i=1}^{n_{\text{trn}}} \mathbf{x}_{\text{trn},i}, \quad \mathbf{c}_{\text{hol}} = \frac{1}{n_{\text{hol}}} \sum_{i=1}^{n_{\text{hol}}} \mathbf{x}_{\text{hol},i}, \quad (9)$$

where $\mathbf{X}_{\text{syn}} \in \mathbb{R}^{n_{\text{syn}} \times d}$ is the matrix with rows representing embeddings of synthetic data, $\mathbf{X}_{\text{trn}} \in \mathbb{R}^{n_{\text{trn}} \times d}$ the matrix for embeddings of training data, and $\mathbf{X}_{\text{hol}} \in \mathbb{R}^{n_{\text{hol}} \times d}$ the matrix for holdout embeddings if provided.

We then compare the centroids of the synthetic and training samples using cosine **similarity**

$$\text{cosine\_similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}, \quad (10)$$

aiming for a high similarity score (with an upper bound of 1). However, to account also here for sampling variance, we use the cosine similarity between the training and holdout centroids as a reference, ensuring that synthetic samples are close to the training distribution without exceeding the similarity expected due to natural sampling noise. To enhance interpretability, we also provide a visualization of the embedded samples and their centroids, projected into a lower-dimensional space using Principal Component Analysis (PCA) (Fig. 7).

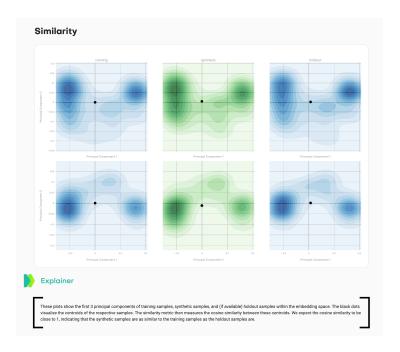[2]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2/

Figure 5: Similarity within PCA-projected embedding space generated by the framework.

In addition to cosine similarity, we leverage the embedding space to train a discriminative model that indicates whether synthetic samples are truly indistinguishable from training samples. If certain properties of the synthetic samples (e.g., implausible attribute combinations) reveal them as synthetic rather than real, the area-under-the-curve (AUC) metric quantifies this distinguishability.

2.3 Distances

Synthetic samples should resemble *novel* samples from the original distribution rather than simply replicating seen samples. Consequently, they are expected to be just as close to training samples as to holdout samples.

Thus, we assess the novelty of synthetic data by examining distances between samples within the high-dimensional embedding space introduced in Section 2.2. For each synthetic sample, we calculate the distance to its closest record (DCR) among the training samples. This nearest-neighbor distance is expected to vary depending on whether the sample is a synthetic inlier or outlier. Therefore, absolute distances alone cannot reliably indicate novelty; instead, we need to contextualize these values by comparing them to the same distances calculated for an equally sized holdout dataset. This comparison is performed for both the average DCR, which we report as a metric, and the overall cumulative DCR distribution, which is visualized (Fig. 6). For reference, the average distances between the synthetic records and their nearest neighbors from

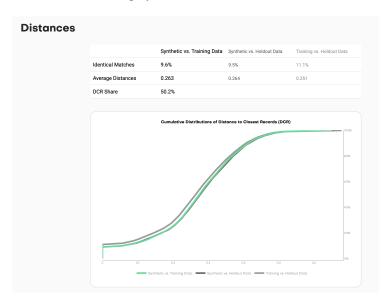the holdout dataset are also displayed.

Distances

	Synthetic vs. Training Data	Synthetic vs. Holdout Data	Training vs. Holdout Data
Identical Matches	9.6%	9.5%	11.1%
Average Distances	0.263	0.264	0.251
DCR Share	50.2%		

Cumulative Distributions of Distance to Closest Records (DCR)

Figure 6: Cumulative distributions of distances to closest records (DCRs) for assessing novelty.

With the sample embeddings denoted as emb_i and i ranging from 1 to N, the nearest neighbor distances are calculated using the L2 norm between embedded representations of synthetic, training, and holdout records. For an embedded synthetic record $\text{emb}_i^{(\text{syn})}$, the distance to its nearest neighbor in the training and holdout datasets is computed as:

$$d_{\text{trn}}^{(i)} = \min_{j \in N_{\text{trn}}} \|\text{emb}_i^{(\text{syn})} - \text{emb}_j^{(\text{trn})}\|_2, \quad d_{\text{hold}}^{(i)} = \min_{j \in N_{\text{hold}}} \|\text{emb}_i^{(\text{syn})} - \text{emb}_j^{(\text{hold})}\|_2. \quad (11)$$

With the indicator function

$$\mathbb{I}_{\text{trn}}^{(i)} = \begin{cases} 1 & \text{if } d_{\text{trn}}^{(i)} < d_{\text{hold}}^{(i)}, \\ 0 & \text{if } d_{\text{trn}}^{(i)} > d_{\text{hold}}^{(i)}, \\ 0.5 & \text{if } d_{\text{trn}}^{(i)} = d_{\text{hold}}^{(i)}, \end{cases} \quad (12)$$

which indicates whether the nearest neighbor of $\text{emb}_i^{(\text{syn})}$ is in the training set, we define the **DCR share** as

$$\text{DCR share} = \frac{1}{N_{\text{syn}}} \sum_{i=1}^{N_{\text{syn}}} \mathbb{I}_{\text{trn}}^{(i)}. \quad (13)$$

It is equally important to compare against the corresponding holdout metrics when evaluating identical matches—instances where synthetic records are exactly the same as the original across all attributes. Crucially, the presence of identical matches does not automatically imply a lack of novelty. If the original data includes duplicates, we should

expect (and even require) a similar level of duplication in the synthetic data. Simply removing individual records in an effort to enforce novelty is not only insufficient but could also increase the risk of exposing original data[14].

3 Empirical Demonstration

By splitting the original data into training and holdout samples and, subsequently, generating multiple synthetic datasets based on the training data, we can effectively compare quality across various generation methods. The chart below visualizes key metrics relative to their holdout-based reference metrics for the UCI Adult Census dataset [15], as synthesized and published in [6]. The closer a synthesizer approaches the *north star* reference point at $(1, 1)$ - the holdout data set - the better its privacy-utility trade-off. As illustrated, this trade-off applies to AI-based data synthesizers just as it does to traditional perturbation techniques. These metrics enable effective comparisons both within and across groups of techniques.
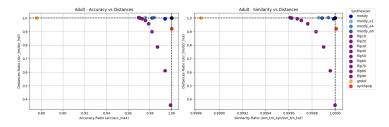

Figure 7: Visualizing the fidelity-privacy tradeoff of different synthesizers using the UCI Adult Census dataset. Accuracy ratio $\text{acc}/\text{acc}_{\text{max}}$ (left) and similarity ratio $\text{sim}_{\text{trn,sim}}/\text{sim}_{\text{trn,hol}}$ (right) across different generative and perturbation techniques. **mostly**: Synthetic data generated using the Synthetic Data SDK [16], with default model and training parameters. **mostly_eX**: same as *mostly* but training is stopped after X epochs. **flipK**: Perturbed dataset where each cell is replaced with a value from a randomly selected record with probability K%. **synthpop** [17] and **gretel** [18]: Other open-source synthesizers.

4 Conclusion

The increasing adoption of generative models for structured data underscores the critical need for interpretable, standardized, and open-sourced tools for synthetic data quality assessment. In response, we have introduced the framework `mostlyai-qa`, a versatile and empirically grounded Python framework that simultaneously quantifies utility and privacy protection of synthetic data. By supporting heterogeneous data structures and providing holdout-based benchmarking, the framework makes it possible to perform comparisons across synthetic data synthesizers and promotes methodological transparency. We anticipate that the framework will support both practitioners and

researchers in the evaluation of synthetic data pipelines, contribute to reproducibility in generative data science, and help to standardize evaluation frameworks in this field.

Acknowledgements

References

[1] Samuel A Assefa, Danial Dervovic, Mahmoud Mahfouz, Robert E Tillman, Prashant Reddy, and Manuela Veloso. Generating synthetic data in finance: opportunities, challenges and pitfalls. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8, 2020.

[2] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. Synthetic data–what, why and how? *arXiv preprint arXiv:2205.03257*, 2022.

[3] Boris van Breugel, Tennison Liu, Dino Oglic, and Mihaela van der Schaar. Synthetic data in biomedicine via generative artificial intelligence. *Nature Reviews Bioengineering*, pages 1–14, 2024.

[4] Bill Howe, Julia Stoyanovich, Haoyue Ping, Bernease Herman, and Matt Gee. Synthetic data for social good. *arXiv preprint arXiv:1710.08874*, 2017.

[5] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-Mu Yu. Empirical evaluation on synthetic data generation with generative adversarial network. In *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*, pages 1–6, 2019.

[6] Michael Platzer and Thomas Reutterer. Holdout-based empirical assessment of mixed-type synthetic data. *Frontiers in big Data*, 4:679939, 2021.

[7] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, Mukund Lahoti, and Pratik Narang. A universal metric for robust evaluation of synthetic tabular data. *IEEE Transactions on Artificial Intelligence*, 5(1):300–309, 2022.

[8] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, Mukund Lahoti, and Pratik Narang. Tabsyndex: A universal metric for robust evaluation of synthetic tabular data. *arXiv preprint arXiv:2207.05295*, 2022.

[9] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pages 290–306. PMLR, 2022.

[10] Erica Espinosa and Alvaro Figueira. On the quality of synthetic generated tabular data. *Mathematics*, 11(15):3278, 2023.

[11] C. Task, K. Bhagat, and G.S. Howarth. SDNist. https://github.com/usnistgov/SDNist, 2023.

[12] Valter Hudovernik, Martin Jurkovič, and Erik Štrumbelj. Benchmarking the fidelity and utility of synthetic relational data. *arXiv preprint arXiv:2410.03411*, 2024.

[13] Reilly Cannon, Nicolette M. Laird, Caesar Vazquez, Andy Lin, Amy Wagler, and Tony Chiang. Assessing generative models for structured data, 2025.

[14] Tobias Hann. Why removing identical matches in synthetic data risks privacy: The swiss cheese problem. pril 2024. Blog post.

[15] Dheeru Dua and Casey Graff. UCI machine learning repository: Adult data set, 2019. University of California, Irvine, School of Information and Computer Sciences.

[16] Mostly AI. Synthetic data sdk. `https://github.com/mostly-ai/mostlyai`.

[17] Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke creation of synthetic data in r. *Journal of statistical software*, 74:1–26, 2016.

[18] Gretel AI. Gretel ai. `https://gretel.ai/`.

A   Summary of Evaluation Metrics

- **Accuracy**: Accuracy is defined as (100% - Total Variation Distance), for each distribution, and then averaged across.

    - `overall`: Overall accuracy of synthetic data, i.e. average across univariate, bivariate and coherence.
    - `univariate`: Average accuracy of discretized univariate distributions.
    - `bivariate`: Average accuracy of discretized bivariate distributions.
    - `coherence`: Average accuracy of discretized coherence distributions. Only applicable for sequential data.
    - `overall_max`: Expected overall accuracy of a same-sized holdout. Serves as reference for `overall`.
    - `univariate_max`: Expected univariate accuracy of a same-sized holdout. Serves as reference for `univariate`.
    - `bivariate_max`: Expected bivariate accuracy of a same-sized holdout. Serves as reference for `bivariate`.
    - `coherence_max`: Expected coherence accuracy of a same-sized holdout. Serves as reference for `coherence`.

- **Similarity**: All similarity metrics are calculated within an embedding space.

    - `cosine_similarity_training_synthetic`: Cosine similarity between training and synthetic centroids.
    - `cosine_similarity_training_holdout`: Cosine similarity between training and holdout centroids. Serves as reference for `cosine_similarity_training_synthetic`.
    - `discriminator_auc_training_synthetic`: Cross-validated AUC of a discriminative model to distinguish between training and synthetic samples.
    - `discriminator_auc_training_holdout`: Cross-validated AUC of a discriminative model to distinguish between training and holdout samples. Serves as reference for `discriminator_auc_training_synthetic`.

- **Distances**: All distance metrics are calculated within an embedding space. An equal number of training and holdout samples is considered.

    - `ims_training`: Share of synthetic samples that are identical to a training sample.
    - `ims_holdout`: Share of synthetic samples that are identical to a holdout sample. Serves as reference for `ims_training`.
    - `dcr_training`: Average L2 nearest-neighbor distance between synthetic and training samples.

- **dcr_holdout**: Average L2 nearest-neighbor distance between synthetic and holdout samples. Serves as reference for dcr_training.
- **dcr_share**: The share of synthetic samples that are closer to a training sample than to a holdout sample. This shall not be significantly larger than 50%.

B Framework Installation and Example Usage

The presented framework for evaluating the quality of synthetic data requires Python version 3.10 or later, and can be easily installed using `pip`:

```
pip install -U mostlyai-qa
```

Once installed, its main interface is the 'report', which expects the data samples to be provided as `pandas` DataFrames:

```python
from mostlyai import qa

# analyze single-table data
report_path, metrics = qa.report(
    syn_tgt_data=synthetic_df,
    trn_tgt_data=training_df,
    hol_tgt_data=holdout_df,
)

# analyze sequential data with context
report_path, metrics = qa.report(
    syn_tgt_data=synthetic_df,
    trn_tgt_data=training_df,
    hol_tgt_data=holdout_df,
    syn_ctx_data=synthetic_context_df,
    trn_ctx_data=training_context_df,
    hol_ctx_data=holdout_context_df,
    ctx_primary_key="id",
    tgt_context_key="user_id",
)
```

Additional usage examples, along with their corresponding HTML reports, are available in the GitHub repository https://github.com/mostly-ai/mostlyai-qa.