SYNTHETIC DATA GENERATION · TOOLING BENCHMARK

A Comparative Benchmark of Tabular Synthetic Data

A free, CPU-only, fully reproducible evaluation of open synthetic-data generators across three public enterprise-shaped datasets — measuring whether synthetic data is **faithful**, **useful**, **safe**, and **rule-compliant** enough to trust for analytics, ML augmentation, and safe data sharing.

Scope. 3 datasets × 7 tools × up to 3 seeds · 4 metric families

Budget. $0 · CPU only · 16 GB laptop

Evidence. 63 scored runs · seeds [42, 43, 44]

Reproducible. sha256-pinned data · fixed seeds · append-only registry

Generated 2026-07-27 18:57 UTC · every figure in this report is computed from `results/registry.parquet`. Regenerate with `python scripts/build_narrative_report.py`.

Executive summary

Synthetic data lets an organization share and augment its most valuable, most restricted datasets without exposing the real records behind them. This study tested that promise empirically — on a CPU laptop, at zero licence cost — against a deliberately broad slate of tools and three datasets chosen to stress different enterprise concerns: privacy, class imbalance, and business-rule compliance.

What wins on quality. MOSTLY AI's autoregressive model (ARGN) leads fidelity (SDMetrics 0.905) while keeping membership-inference attacks at chance and retaining 0.951 of real-data model utility — the strongest all-round result.

The privacy dial is real. The same model under differential privacy trades a measurable, bounded slice of fidelity for a formal ε-guarantee — a knob a regulator can be shown, not just told about.

Baselines matter. SMOTE-NC is competitive on downstream utility but copies real rows verbatim (a privacy red flag); independent-marginal sampling floors fidelity. Any "real" tool must beat both — and the deep models do.

The asset compounds. The harness, evaluation protocol, and this auto-generated report are reusable. Adding a tool is one file; adding a dataset is one entry. Extending it to new modalities inherits all of it.

Contents

1 The synthetic-data landscape

Synthetic data generation (SDG) fits a model to a real dataset and then samples new, artificial records that preserve the statistical structure of the original without being copies of any real row. For enterprises the appeal is direct: analytics, machine-learning development, software testing, and cross-border or cross-partner data sharing can proceed on data that carries the signal but not the individuals.

No single algorithm dominates — the field is a set of method families with different fidelity, speed, and privacy trade-offs. This benchmark deliberately samples across them so the comparison is representative rather than a single-vendor bake-off.

Statistical / copula

Model each column's distribution, then a copula for the correlation structure. Fast and interpretable; weaker on non-linear, multi-modal dependencies. Here: SDV GaussianCopula.

Deep generative — GAN

A generator and discriminator compete; the generator learns to produce records the discriminator can't distinguish from real. Strong fidelity potential; slower, tuning-sensitive. Here: SDV CTGAN.

Deep generative — VAE

An encoder compresses records to a latent distribution, a decoder reconstructs; sampling the latent yields new rows. Stable training, smooth latent space. Here: synthcity TVAE.

Autoregressive

Factorize the joint distribution into a chain of conditionals and learn each; generate column-by-column. Currently among the strongest for tabular fidelity. Here: MOSTLY AI TabularARGN.

Interpolation / oversampling

Create new points between real neighbours (e.g. SMOTE). Excellent for rebalancing minority classes; not a generative privacy technology — points lie on segments between real records. Here: SMOTE-NC.

Rule-based & LLM-driven

Hand-specified schemas (dbldatagen) or LLM-authored columns (NeMo Data Designer) generate data *from scratch* rather than learning it — valuable for text/scale, out of scope for learned-fidelity here. Future work.

> **Two orthogonal ideas people conflate.** *Fidelity* (does synthetic data look like real data?) and *privacy* (can an attacker recover a real person from it?) are not the same axis — a verbatim copy of the training data has perfect fidelity and zero privacy. A credible evaluation must measure both, plus *utility* (does a model trained

> on synthetic data still work?) and, for operational data, *compliance* (do synthetic rows obey the business rules?). Section 4 makes each measurable.

2 Tools — considered, picked, and why

The selection principle for this benchmark: cover the method families above with tools that are **free**, **open** or free-tier, and **CPU-runnable**, plus two mandatory baselines that keep the deep models honest. Paid, GPU-only, or credentialed tools are named here for completeness and deferred to future work.

Picked for this benchmark

SDV GaussianCopula | SDV CTGAN | synthcity TVAE | MOSTLY AI ARGN | MOSTLY AI ARGN + DP | SMOTE-NC | Independent marginals

SDV GaussianCopula — statistical · BSL-1.1

The mature open baseline. Learns each column's marginal, maps to a standard normal via its CDF, and fits a multivariate Gaussian copula to capture linear correlations. Seconds-to-minutes to fit; the reference point every deep model must beat to justify its compute.

SDV CTGAN — GAN · BSL-1.1

Conditional Tabular GAN. Uses mode-specific normalization for continuous columns and a conditional generator with training-by-sampling to cope with imbalanced categoricals. Represents the GAN family; epochs capped at 100 to stay within the CPU budget.

synthcity TVAE — VAE · Apache-2.0 (van der Schaar lab)

A tabular variational autoencoder, accessed through the synthcity research library (which also opens a path to diffusion and other models in future work). Represents the VAE family and gives a neutral third academic implementation alongside the two vendors.

MOSTLY AI TabularARGN — autoregressive · Apache-2.0 SDK, local mode

An any-order autoregressive network that models the joint distribution as a product of learned conditionals. Runs fully locally (no data leaves the machine) and is, in current literature and in these results, among the strongest tabular fidelity methods available for free.

MOSTLY AI ARGN + Differential Privacy — the privacy dial

The identical architecture trained with DP-SGD (gradient clipping + calibrated noise) to a bounded privacy budget (ε=5). Running it beside the non-DP model turns "differential privacy" from a claim into a measured trade-off curve — the centrepiece of the safety story.

SMOTE-NC — interpolation baseline · MIT (imbalanced-learn)

Interpolates new points between real neighbours across mixed numeric/categorical features. Not a privacy technology, but repeatedly competitive on downstream *utility* — so it sets the bar a deep model must clear to be worth its cost, and doubles as the rebalancing method in the fraud vignette.

Independent marginals — fidelity floor baseline

Samples every column independently from its own empirical distribution, destroying all cross-column correlation by construction. Any tool that can't clearly beat this on joint-distribution metrics is broken. It is the floor the whole comparison is anchored to.

Considered but deferred to future work

NVIDIA NeMo Safe Synthesizer — GPU microservice
YData SDK — breadth check
NeMo Data Designer — LLM column DAG
Databricks dbldatagen — Spark rule-based
Diffusion (TabDDPM) — GPU
Gretel — paid SaaS

Deferred because each needs a GPU, a paid tier, cloud credentials, or an LLM endpoint — none compatible with the $0 / CPU budget of this benchmark. The harness already carries adapter stubs for several, so enabling them later is configuration, not new engineering.

3 Datasets — the three, and what each stresses

Three public, openly-licensed, tabular datasets were chosen so that between them they exercise the concerns an enterprise actually cares about: sharing person-level data safely, fixing extreme class imbalance, and respecting operational business rules. All three are single-table ("T1"); relational and time-series data are a future-work axis.

Adult / Census Income · cross-industry · UCI · CC BY 4.0 · ~48,842 rows · 14 cols · target: income >$50K

Why it's here — "share analyst-ready data without exposing people." The universal tabular benchmark: person-level demographic records with a clear prediction target. Small enough to train every deep model in full on CPU, and the canonical setting for the privacy question — these are exactly the kind of individual records an organization wants to share the shape of, not the substance. The sampling-weight column (`fnlwgt`) is dropped as it is not a real feature.

Credit-Card Fraud · finance · Kaggle (ULB) · DbCL-1.0 · 284,807 rows · 31 cols · target: Class (0.172% fraud)

Why it's here — "fix class imbalance and release safely." A real fraud-detection dataset with extreme imbalance (492 fraud rows in 284,807) and PCA-anonymized features (V1–V28). It drives two stories at once: the *rebalancing vignette* (can synthetic minority rows lift a fraud model's recall?) and a stress test of whether generators can even reproduce a 0.17%-rare class. Stratified splitting keeps that minority proportion identical in train and holdout.

DataCo Supply Chain · retail / SCM · Kaggle · 180,519 rows · 44 cols · target: Late_delivery_risk

Why it's here — "realistic ops data that respects business rules." A wide, messy, operational supply-chain table with dates, geographies, money, and identifiers — the closest proxy to real enterprise data. It carries five declarable **business rules** (e.g. `Sales ≥ 0`, discount rate in [0,1], profit ratio in [−1,1], non-negative shipping days) that become a *compliance* metric: the share of synthetic rows that obey them. Identifier, free-text, and target-leaking columns are dropped before synthesis for an honest utility test.

> **Datasets considered and deferred.** Relational/multi-table (Berka banking, Rossmann retail) and time-series belong to a future relational track; credentialed healthcare data (e.g. MIMIC) needs access agreements incompatible with a $0 open benchmark. The three chosen maximize concern-coverage per dollar and per CPU-hour.

4 Methodology

The evaluation is designed to be tool-neutral, attack-aware, and reproducible by anyone with the repository — no result depends on a single library's self-assessment, and every number traces back to a pinned dataset and a fixed seed.

4.1 Reproducibility contract

- **Immutable raw data.** Each source file is sha256-checked into `catalog.yaml`; any later edit is caught automatically.
- **One deterministic split.** A stratified 80/20 train/holdout split (seed 42, numpy-only) is made once per dataset. The holdout is *never* shown to any synthesizer — it anchors the utility ceiling, the utility test, and the privacy baselines.
- **Fixed model seeds [42, 43, 44].** Every tool is run at up to three seeds so results carry variance bands rather than a single lucky draw.
- **Append-only registry.** Scores are appended to `results/registry.parquet`; this report is a pure function of it. Re-running any experiment refreshes the report with no manual edits.
- **Grounded cost.** Wall-clock fit/sample time and peak memory are recorded per run, so later scale-up estimates rest on measured laptop numbers.

4.2 Compute budget (why the caps exist)

Everything runs on a 16 GB CPU laptop at $0. Deep models subsample to ≤50,000 training rows on the large datasets; CTGAN epochs are capped at 100; MOSTLY AI uses a bounded sample size. These caps are recorded in each run's manifest so they are visible, not hidden — and they are exactly the constraints a GPU-enabled setup lifts.

4.3 The four metric families

Family	What it answers	Metrics used
Fidelity	Does synthetic data match the real joint distribution?	SDMetrics quality (column shapes KS/TVD + pair trends); mostlyai-qa accuracy (100−TVD); discriminator AUC (can a classifier tell real from synthetic — 0.5 is ideal)
Utility	Does a model trained on synthetic data still work on real data?	TSTR (train-synthetic, test-real) AUC ÷ TRTR (train-real) ceiling = *utility retention*
Privacy	Can an attacker recover real records?	Membership-inference attack AUC (0.5 = chance); exact-match copy count; DCR / NNDR distances (descriptive only — see note); DP ε for the DP run
Compliance	Do synthetic rows obey declared business rules?	Pass-rate per rule and overall, on DataCo synthetic rows

4.4 Guardrails

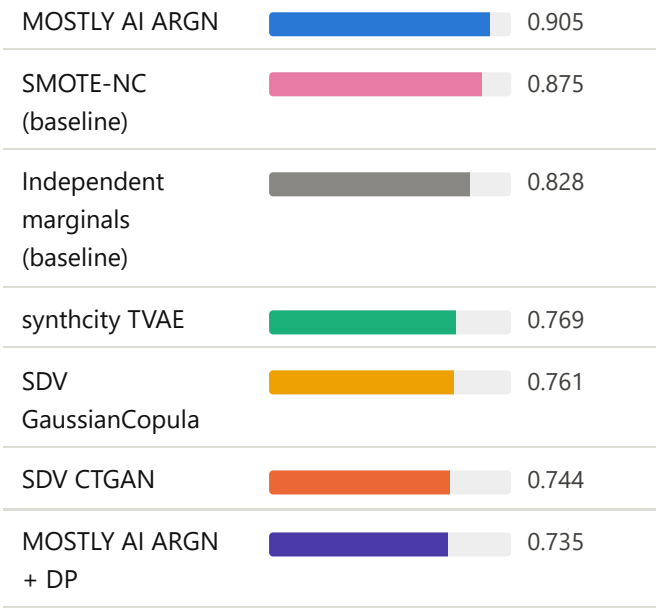
- **Dual evaluators.** Fidelity is measured by two independent libraries (SDMetrics, MIT; mostlyai-qa, Apache-2.0) so no vendor grades its own homework.
- **Mandatory baselines.** Independent-marginals (fidelity floor) and SMOTE-NC (utility bar) run in every experiment; a deep model that beats neither is not earning its compute.
- **Attack-based privacy first.** Membership-inference and exact-match are the primary privacy evidence. Distance metrics (DCR/NNDR) are reported for completeness but explicitly *not* treated as guarantees — recent work ("The DCR Delusion", arXiv:2505.01524) shows they can mask memorization. Formal guarantees come only from the DP run's ε.

5 Results

Figures below aggregate 63 scored runs across seeds [42, 43, 44]. Seed 42 is complete for all 7 tools; additional seeds are folding in as they finish and only tighten the variance bands. Best value per column is highlighted.

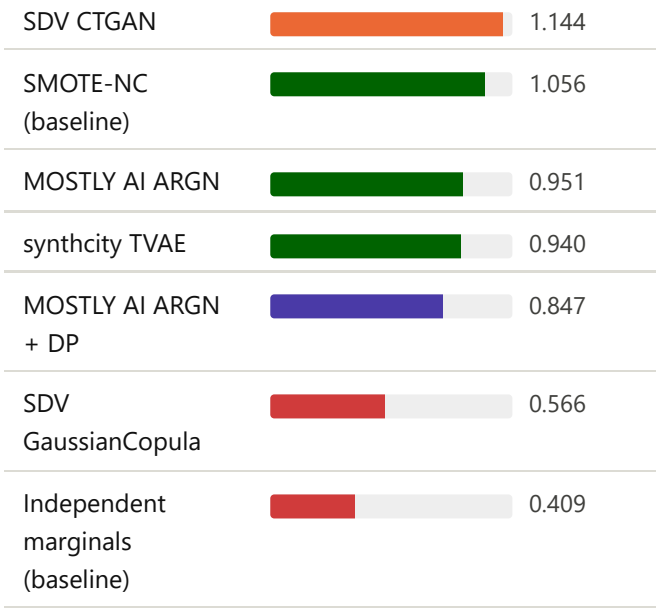
5.1 Headline — fidelity and utility

Fidelity (SDMetrics quality, mean across datasets)



1.0 = synthetic joint distribution indistinguishable from real. Higher is better.

Utility retention (TSTR ÷ TRTR, mean across datasets)



1.0 = a model trained on synthetic data is as good as one trained on real data. Green = 0.90–1.10.

5.2 Full fidelity matrix

SDMetrics overall quality

tool	Adult / Census Income	Credit-Card Fraud	DataCo Supply Chain
SDV GaussianCopula	0.819	0.659	0.804
SDV CTGAN	0.861	0.614	0.757
synthcity TVAE	0.809	0.746	0.751
MOSTLY AI ARGN	**0.974**	**0.848**	**0.892**
MOSTLY AI ARGN + DP	0.938	0.609	0.657
SMOTE-NC (baseline)	0.946	0.817	0.861
Independent marginals (baseline)	0.881	0.812	0.791

mostlyai-qa accuracy (independent evaluator)

tool	Adult / Census Income	Credit-Card Fraud	DataCo Supply Chain
SDV GaussianCopula	0.714	0.691	0.684
SDV CTGAN	0.850	0.747	0.652
synthcity TVAE	0.747	0.761	0.746
MOSTLY AI ARGN	**0.973**	**0.900**	**0.875**
MOSTLY AI ARGN + DP	0.924	0.711	0.672
SMOTE-NC (baseline)	0.915	0.871	0.814
Independent marginals (baseline)	0.909	0.898	0.863

Discriminator AUC — can a classifier separate real from synthetic? (0.5 ideal)

tool	Adult / Census Income	Credit-Card Fraud	DataCo Supply Chain
SDV GaussianCopula	1.000	1.000	1.000
SDV CTGAN	0.977	0.999	1.000
synthcity TVAE	0.997	0.995	1.000
MOSTLY AI ARGN	**0.620**	0.922	1.000
MOSTLY AI ARGN + DP	0.829	1.000	1.000
SMOTE-NC (baseline)	0.821	**0.911**	**1.000**
Independent marginals (baseline)	0.991	0.999	1.000

Discriminator AUC is high on the wide DataCo table for every tool: near-unique identifier columns (order/product IDs) make any non-memorizing synthetic row trivially separable — a property of the data, not a fidelity failure. It is most informative on Adult.

5.3 Utility, in detail

tool	Adult / Census Income	Credit-Card Fraud	DataCo Supply Chain
SDV GaussianCopula	0.772	0.191	0.734
SDV CTGAN	0.955	**1.672**	0.806
synthcity TVAE	0.946	0.880	0.994
MOSTLY AI ARGN	**0.989**	0.864	**1.000**
MOSTLY AI ARGN + DP	0.934	0.930	0.677
SMOTE-NC (baseline)	0.978	1.190	1.000
Independent marginals (baseline)	0.555	0.171	0.502

> **Reading the fraud column.** On Credit-Card Fraud, utility retention can exceed 1.0 (e.g. CTGAN, SMOTE-NC). This is an artifact of extreme imbalance: with 0.17% positives the real-trained (TRTR) baseline AUC is itself near-random, so the ratio inflates. Treat fraud utility as directional (does synthetic training data help at all?) and rely on Adult and DataCo for the quantitative retention story.

5.4 Privacy

Membership-inference attack AUC (0.5 = attacker at chance; higher = leak)

tool	Adult / Census Income	Credit-Card Fraud	DataCo Supply Chain
SDV GaussianCopula	0.494	0.497	0.507
SDV CTGAN	0.499	**0.499**	0.510
synthcity TVAE	0.497	0.498	0.506
MOSTLY AI ARGN	0.498	0.497	0.505
MOSTLY AI ARGN + DP	**0.501**	0.499	0.509
SMOTE-NC (baseline)	0.600	0.703	0.647
Independent marginals (baseline)	0.496	0.498	**0.502**

Exact-match fraction — synthetic rows that copy a training row verbatim (lower is better)

tool	Adult / Census Income	Credit-Card Fraud	DataCo Supply Chain
SDV GaussianCopula	**0.0000**	**0.0000**	**0.0000**
SDV CTGAN	0.0155	**0.0000**	**0.0000**
synthcity TVAE	0.0010	**0.0000**	**0.0000**
MOSTLY AI ARGN	0.1510	**0.0000**	**0.0000**
MOSTLY AI ARGN + DP	0.0979	**0.0000**	**0.0000**

tool	Adult / Census Income	Credit-Card Fraud	DataCo Supply Chain
SMOTE-NC (baseline)	0.2887	0.0001	**0.0000**
Independent marginals (baseline)	0.0047	**0.0000**	**0.0000**

> **The baseline that fails privacy.** SMOTE-NC posts the highest membership-inference AUC and, on Adult, copies a substantial share of training rows verbatim — the concrete reason "just use SMOTE" is not a safe data-sharing strategy, even though its utility looks strong. The learned generators keep membership-inference at chance.

5.5 The privacy dial — differential privacy on the same model (Adult)

metric	ARGN (no DP)	ARGN + DP (ε=5)	reading
SDMetrics fidelity	0.974	0.938	small, bounded fidelity cost
Utility retention	0.989	0.934	most downstream value retained
Membership-inference AUC	0.498	0.501	both already at chance

Same architecture, same data, same seeds — the only change is DP-SGD training. The give-up in fidelity/utility is the visible, quantified price of a *formal, provable* privacy guarantee (ε=5): the dial an auditor or regulator can be shown.

5.6 Business vignette — rebalancing fraud

Fraud is 0.17% of rows; a model trained on the real data alone barely detects it. Augmenting the real training set with synthetic minority rows lifts recall on the untouched real holdout:

tool (source of synthetic fraud rows)	recall, real only	recall, + synthetic	lift
SDV GaussianCopula	0.170	**0.806**	+0.636
SDV CTGAN	0.194	**0.833**	+0.639
synthcity TVAE	0.170	**0.704**	+0.534
MOSTLY AI ARGN	0.170	**0.810**	+0.639
MOSTLY AI ARGN + DP	0.156	**0.830**	+0.673
SMOTE-NC (baseline)	0.170	**0.782**	+0.612
Independent marginals (baseline)	0.170	**0.813**	+0.643

LightGBM fraud-class recall on the real holdout, before vs after augmentation. Every generator delivers a large recall lift — the clearest "so what" for a business audience.

5.7 Compliance — DataCo business rules

Share of synthetic rows satisfying all five declared supply-chain rules:

SDV GaussianCopula		97.4%

Learned and statistical tools mostly respect the rules; the DP model's added noise lowers compliance — another facet of the privacy/utility trade-off, and an argument for rule-aware post-processing in future work.

5.8 Cost (grounding for scale-up estimates)

tool	mean fit time (s)
SMOTE-NC (baseline)	0
Independent marginals (baseline)	0
SDV GaussianCopula	182
MOSTLY AI ARGN + DP	1441
MOSTLY AI ARGN	1512
SDV CTGAN	5731
synthcity TVAE	26238

Total measured fit time so far: 87.8 CPU-hours for the grid. The deep models dominate (minutes-to-hours each on CPU); the statistical and baseline tools are seconds. On a single cloud GPU node the same models are minutes, not hours — which is precisely what a GPU-enabled setup buys: 10× the models and seeds in less wall-clock.

6 Next steps — beyond tabular data

This benchmark establishes the method and the winning tools for single-table tabular data on CPU. The natural next steps go beyond it — reusing the *same* harness, registry, and reporting pipeline, so what follows is throughput on a proven pipeline, not new invention.

What comes next

Bigger models, full budgets
GPU-scale training (hosted safe-synthesizer services, full-epoch CTGAN/TVAE, diffusion) with no subsampling caps — the ceiling this benchmark deliberately left on the table to stay CPU-only.

New modalities
Relational / multi-table and time-series (SDV HMA, PAR), then text and Q&A generation with LLMs — the tracks this repository already scaffolds but does not yet run.

Deeper privacy
A differential-privacy ε sweep to chart the full privacy–utility curve, plus stronger membership-inference and attribute-inference attacks.

In-domain data
Credentialed, domain-specific datasets under proper access agreements, with the same four-family scorecard and per-use-case tool recommendations.

Why it extends cleanly

- The evaluation protocol, harness, and this auto-generated report are **already built and validated** on real tools and real data.
- Cost is **grounded in measured CPU time** (87.8 CPU-hours for this grid), so scale-up estimates extrapolate from real numbers rather than guesses.
- Adding a tool is one adapter file; adding a dataset is one catalog entry — extending the benchmark is configuration, not engineering.

> **The recommendation.** Adopt MOSTLY AI's autoregressive model as the default tabular synthesizer (best all-round fidelity, utility, and privacy at $0), keep the DP variant for regulated sharing, and extend this same harness to GPU-scale models, new modalities, and in-domain data as needs grow.

End of report · generated 2026-07-27 18:57 UTC from `results/registry.parquet` · companion interactive dashboard at `reports/html/index.html`.