STAT414 Final Report

Predictive & Generative AI for Digital Marketing Data

Hochan Son, Stephanie Lu, Ashwin Ramaseshan,
hochanson@ucla.edu, stephaniehlu@g.ucla.edu, arama035@g.ucla.edu
and Juyi Yang
juyiyang30@g.ucla.edu

University of California, Los Angeles

March 23, 2025

Abstract

Digital marketing relies increasingly on machine learning models to optimize campaign performance, target relevant audiences, and predict consumer behavior. However, these models face persistent challenges: extreme class imbalance in click prediction (with positive examples often representing less than 2% of observations), privacy restrictions limiting data sharing, and the high cost of obtaining labeled examples. Synthetic data generation has emerged as a promising solution to these challenges, enabling marketers to augment training datasets with artificially created examples that maintain statistical properties of real data while addressing imbalance and privacy concerns. This study investigates the effectiveness of five state-of-the-art synthetic data generation approaches for digital marketing applications, with particular focus on their ability to preserve both statistical fidelity and predictive utility. Generative AI models like CTGAN, TabDDPM [2], and TVAE [3] address challenges in synthetic tabular data generation, including class imbalance, privacy, and distribution fidelity. CTGAN outperforms others in capturing realistic distributions and improving predictive accuracy through minority-class oversampling SMOTE. And, LLMs like GPT-2 based Tabular DATA Generator use structured tokenization for high statistical fidelity, while TVAE struggles with distributional similarity despite moderate success in downstream tasks.

Predictive & Generative AI for Digital Marketing Data

Hochan Son

Stephanie Lu

Ashwin Ramaseshan

Juyi Yang

March 23, 2025

1 Introduction

Synthetic data generation has emerged as a vital solution to address challenges such as data scarcity, imbalance, and privacy concerns in real-world datasets. This study focuses on leveraging generative AI to produce synthetic digital marketing data aimed at improving ad targeting, campaign optimization, and model generalization. By generating high-quality synthetic data that mimics real-world statistical properties, this research seeks to enhance the utility of machine learning models while maintaining privacy and addressing class imbalance issues. Class imbalance is particularly problematic in advertising click prediction, where positive examples (clicks) are significantly outnumbered by negative examples (non-clicks). Traditional methods like SMOTE, random oversampling, and undersampling struggle to handle high-dimensional data with complex categorical features. Conditional Tabular Generative Adversarial Networks (CTGAN) overcome these limitations by generating synthetic samples that preserve intricate relationships between user attributes, ad characteristics, and click behavior. This ensures that the generated data reflects the true distribution of the original dataset, enabling the development of more robust prediction models [5]. Recent advancements in transformer-based language models offer a novel approach to synthetic data generation. These models leverage self-attention mechanisms to capture complex dependencies between variables that traditional statistical methods fail to model effectively. This paper introduces TabularGeneratorGPT2(TG-GPT2), a framework that repurposes generative pre-trained transformers for tabular data synthesis. By employing a specialized tokenization strategy that encodes tabular structures with markers for rows, columns, and data types, the framework generates synthetic records that maintain statistical fidelity while introducing sufficient variance for privacy [6]. Similarly, Tabular Variational Autoencoders

(TVAE) provides another approach by using variational inference to encode relationships between numerical and categorical features. This study evaluates TVAE's ability to generate advertising data that preserves statistical properties while ensuring diversity and privacy through preprocessing techniques like normalization and factorization [7].

2 Methodology

There are various well-known methods among many effective Generative methods. We were particularly drawn to a few AI models, which have been covered in the class, such as CTGAN, TVAE, LLMs(TabularGenerator-GPT2), and TabDDPM. Also, we provided a balanced predictive algorithm to compare the results with SMOTE. To give a glance at a summary of the benefit of those Generative AI models,

- **SMOTE**: A synthetic minority over-sampling technique that creates new instances by interpolating between existing minority class examples. Unlike random oversampling, SMOTE generates synthetic examples along the line segments joining neighboring minority class instances, helping to avoid overfitting.
- **CTGAN**: A GAN designed specifically for tabular data, simultaneously training a generator and discriminator.
- **TVAE**: Utilizes variational autoencoders tailored for tabular data, capturing complex feature dependencies.
- **LLMs (TabularGenerator-GPT2)**: Fine-tuned language models based on GPT-2 to adapt tabular data format generation.
- **tabddpm**: Employs diffusion processes to iteratively generate realistic tabular data.

2.1 Dataset Description

This study utilized a digital marketing dataset consisting of 7 million records from the marketing ad click dataset, containing user interactions with online advertisements. The dataset includes 35 features covering user demographics (age, gender), device information (device type, screen size), contextual data (time of day, website category), and advertisement attributes (creative type, position). The target variable represents click events, with an extreme imbalance ratio of 99:1 (non-clicks to clicks).

The complete dataset contained approximately over 7 million records. Due to computational constraints, we sampled 10

3 Overview of models

3.1 CTGAN

Our approach employs Conditional Tabular Generative Adversarial Networks (CTGAN), which are specifically designed for generating synthetic tabular data with mixed categorical

and continuous features. The CTGAN architecture consists of two competing networks: a generator G and a discriminator D. The generator attempts to create synthetic samples that resemble real data, while the discriminator tries to distinguish between real and synthetic samples. Both networks improve iteratively through adversarial training. Formally, the CTGAN optimization objective can be expressed as:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

where $p_{data}(x)$ represents the distribution of real data, and $p_z(z)$ is the prior distribution of the latent space. To address the class imbalance, we focus on generating synthetic samples specifically for the minority class. Given an imbalanced dataset with majority class samples X_{maj} and minority class samples X_{min}, we first train the CTGAN model on X_{min}, then generate synthetic samples X_{syn} to achieve a target ratio r between classes:

$$|X_{syn}| = |X_{maj}| \cdot (1/r - 1) - |X_{min}| \tag{2}$$

where $|X|$ denotes the cardinality of set X. This approach maintains the original majority class distribution while augmenting the minority class to the desired proportion.

3.2 TVAE

The TVAE framework implements a probabilistic encoder-decoder architecture that maps input data $\mathbf{x}$ to a lower-dimensional latent space $\mathbf{z}$ and then reconstructs it. The encoder network parameterized a variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$ as a multivariate Gaussian with mean $\boldsymbol{\mu}_\phi(\mathbf{x})$ and diagonal covariance $\boldsymbol{\sigma}^2_\phi(\mathbf{x})$, while the decoder network represents the likelihood $p_\theta(\mathbf{x}|\mathbf{z})$. The model is trained by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{3}$$

where D_{KL} is the Kullback-Leibler divergence that regularizes the latent space. For our advertising data, we employed a Mean Squared Error (MSE) loss for reconstruction, optimizing with Adam (learning rate = 0.001) for 50 epochs. The synthetic data generation strategy was designed to maintain class balance by generating 30% synthetic samples for the dominant class and 100,000 samples for the secondary class, ensuring all synthetic labels were correctly assigned as either 0 or 1.

3.3 TabularGenerator-GPT2(LLMs)

The core innovation of our approach lies in the specialized tokenization strategy for representing tabular data. Each row is enclosed within `<ROW>` and `<ENDROW>` tokens, while columns are delimited by `<COL>` and `<ENDCOL>` tokens. To preserve type information, we introduce type-specific tokens: `<NUM>` for numerical values, `<CAT>` for categorical values, and `<LIST>` for list-type values. Formally, a table T with m rows and n columns is transformed into a sequence of tokens where each row r_i is represented as:

$$r_i = \{\texttt{<ROW>}, c_{i,1}, c_{i,2}, \ldots, c_{i,n}, \texttt{<ENDROW>}\} \tag{4}$$

where each column entry $c_{i,j}$ is represented as:

$$c_{i,j} = \{\texttt{<COL>}, \text{name}_j, \texttt{<TYPE>}, \text{value}_{i,j}, \texttt{<ENDCOL>}\} \tag{5}$$

with <TYPE> being one of {<NUM>, <CAT>, <LIST>}. This representation is then used to fine-tune a GPT-2 model using a standard language modeling objective, minimizing the negative log-likelihood,

$$\mathcal{L} = -\sum_{t=1}^{l} \log P(s_t | s_{<t}) \tag{6}$$

where s_t is the token at position t and $s_{<t}$ represents all previous tokens. For synthetic data generation, we start with a <ROW> token and sample subsequent tokens autoregressively using temperature sampling ($T = 0.8$) to balance faithfulness to the learned distribution and diversity of generated samples.

3.4 TabDDPM

TabDDPM is a novel approach to generating realistic tabular data by adapting diffusion probabilistic models (originally effective in image and time-series domains) to mixed data types (numerical and categorical). It systematically adds noise to real data during a *forward process* and trains a *reverse process* (i.e., denoising) network to remove this noise step-by-step. By learning the underlying distribution of tabular features, TabDDPM produces synthetic tables that closely mirror the real data distribution while capturing complex dependencies among columns.

3.4.1 Background on Diffusion model

Let $\mathbf{x}_0$ be drawn from the real dataset distribution $q(\mathbf{x}_0)$. The forward diffusion process produces latent variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\Big(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\, \beta_t \mathbf{I}\Big), \tag{7}$$

where β_t is a variance-scheduling parameter controlling noise at time t. The reverse process defines:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \approx \mathcal{N}\Big(\mathbf{x}_{t-1};\, \boldsymbol{\mu}_\theta(\mathbf{x}_t,\, t),\, \boldsymbol{\Sigma}_\theta(\mathbf{x}_t,\, t)\Big), \tag{8}$$

where $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are predicted by a neural network parameterized by θ. During training, the model aligns these estimates to invert the forward noising process.

3.4.2 Adapting Diffusion to Tabular Data

Tabular datasets commonly combine *continuous* (real-valued) and *categorical* features. TabDDPM handles this heterogeneity via:

Mixed Feature Representation. Continuous features remain in $\mathbb{R}$, following the original Gaussian noising scheme. Categorical features are embedded into continuous vectors, so they can be perturbed by the same Gaussian diffusion mechanism.

Forward Process for Categorical Data. Rather than injecting noise into raw discrete values, TabDDPM injects Gaussian noise into the *embedding space*. This continuous embedding is learned and preserves semantic information of categorical features.

3.4.3 Architecture and Training

Network Architecture. A U-Net-like or Transformer-based model predicts $(\boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$ from the current noisy sample $\mathbf{x}_t$ and a learned time-embedding of t. The network output is typically parameterized as a mean plus (possibly) a predicted or fixed variance.

Loss Function. The training follows the original DDPM framework:

$$L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\big[\, \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon},\, t)\|^2 \big], \tag{9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. For continuous embeddings of categorical columns, one may use MSE-style losses or cross-entropy objectives at the final projection layer.

Sampling. To sample a synthetic point, one starts with Gaussian noise $\mathbf{x}_T$ and iteratively draws

$$\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \quad \text{for } t = T, \ldots, 1. \tag{10}$$

After reaching $\mathbf{x}_0$, continuous columns remain in $\mathbb{R}$; embeddings for categorical columns are decoded (e.g., by choosing the nearest or highest-likelihood discrete label).

4 Implementation

4.1 CTGAN

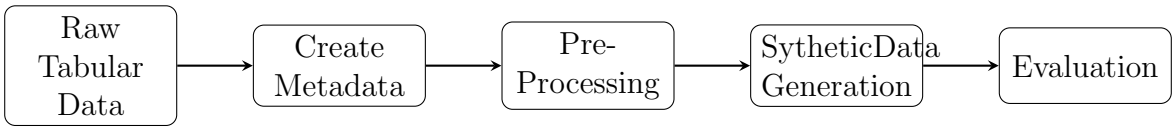


Figure 1: simplified CTGAN system architecture.

The implementation consists of a three-phase approach to synthetic data generation and model evaluation, as illustrated in Figure 1. In the first phase, we preprocess the advertising click data, addressing special format columns (such as caret-separated lists) and applying feature engineering techniques including one-hot encoding for low-cardinality categorical features and target encoding for high-cardinality categorical features. The second phase involves synthetic data generation using three distinct approaches: (1) pure CTGAN-based minority class augmentation, (2) traditional SMOTE oversampling, and (3) a hybrid approach combining CTGAN and SMOTETomek. We containerized the implementation using Docker to ensure reproducibility and platform independence, particularly important given library compatibility challenges with different operating systems. The third phase involves training machine learning models on the original imbalanced data and the three synthetic datasets,

followed by comprehensive evaluation. To address practical considerations of dataset size, we implemented sampling parameters to work with 10% of the original data and imposed a maximum limit of 100,000 records for both training and generation. For quality assurance, we incorporated validation metrics to evaluate the synthetic data, including distribution comparison of numerical features, correlation preservation, dimensionality reduction visualization, and privacy leakage analysis to ensure the synthetic data doesn't inadvertently expose sensitive information from the original dataset.

4.2 TVAE

We implemented TVAE with a symmetric encoder-decoder architecture using PyTorch. The encoder consists of two hidden layers with 32 units each, followed by a latent space projection to 16 dimensions. The decoder mirrors this structure to reconstruct the original feature space from the latent representation. Both networks utilize ReLU activation functions for the hidden layers, with the encoder outputting both mean and variance parameters for the latent distribution.

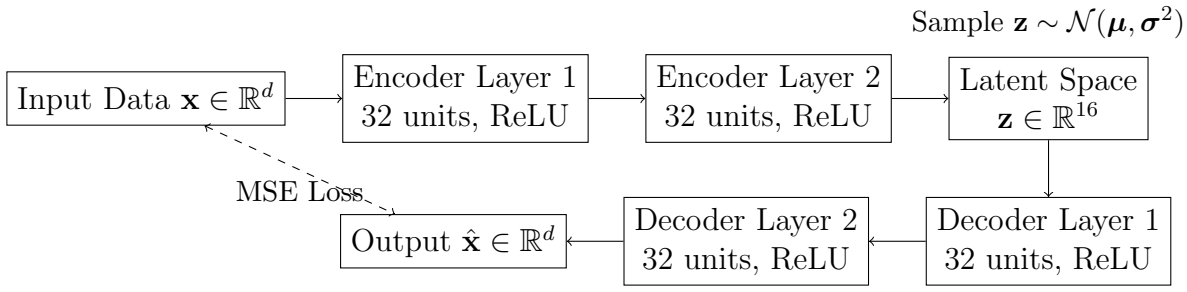


Figure 2: TVAE Architecture for Synthetic Advertising Data Generation

4.3 TabularGenerator-GPT2(LLMs)

The TabularGenerator-GPT2(TG-GPT2) is implemented as a Python class with a modular architecture that handles data preprocessing, model training, synthetic data generation, and quality evaluation. Figure 3 illustrates the overall system architecture, highlighting the flow of data through the different components. The framework begins with a data loading phase that infers column types and prepares the dataset for tokenization. The tokenization module then converts the tabular data into the specialized token format described in the methodology section. The resulting token sequences are used to fine-tune a pre-trained GPT-2 model, which learns both the distribution of individual columns and the relationships between them.

The implementation includes several key features to enhance usability and performance. First, it dynamically adapts to different computational environments by adjusting training parameters based on available hardware. For high-performance environments with GPUs, it uses larger batch sizes (4) and higher learning rates (5e-5), while for environments with limited resources, it reduces these parameters to ensure stable training. Second, the framework implements extensive post-processing to ensure the generated data adheres to the expected

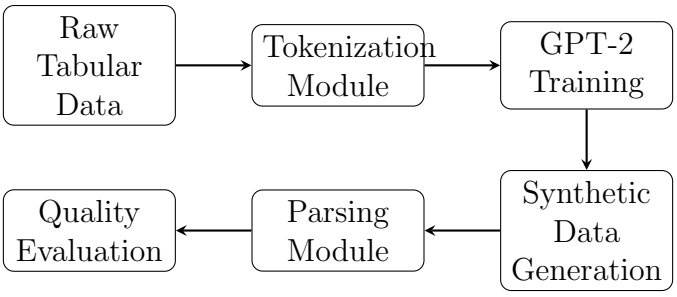


Figure 3: simplified TG-GPT2 system architecture

formats and ranges, including proper handling of numerical values, categorical distributions, and list-type fields. Finally, it provides comprehensive evaluation metrics to assess the quality of the generated data, measuring statistical fidelity, distributional similarity, and correlation preservation between the original and synthetic datasets.

4.3.1 Model Architecture

Our framework utilizes the GPT-2 model architecture, a transformer-based language model with a decoder-only structure. The model consists of multiple transformer blocks, each containing a multi-headed self-attention mechanism and a feed-forward neural network. For a given input sequence of tokens, the model predicts the next token in the sequence, which we leverage to generate synthetic tabular data.

The standard GPT-2 architecture is augmented with additional vocabulary tokens for our tabular data representation. Formally, the model computes:

$$P(s_t|s_{<t}) = \text{softmax}(h_t W^T) \tag{11}$$

where h_t is the final hidden state corresponding to the t-th token, and W is the token embedding matrix.

4.3.2 Tokenization

The tokenization strategy employed in this model offers several significant advantages that work together to enable effective tabular data modeling. By preserving the inherent tabular structure within a sequential format that transformers can process, this approach allows the model to understand the two-dimensional nature of data tables while working within the constraints of sequence-based architectures. The explicit encoding of column names maintains the crucial relationship between field identifiers and their values, ensuring semantic consistency across generated records. Furthermore, the differentiation between data types through specialized tokens helps the model learn appropriate statistical distributions and patterns specific to numerical, categorical, and list-type values. Perhaps most importantly, this structured tokenization creates a consistent grammatical framework that the model can internalize during training and then faithfully reproduce when generating new synthetic data, resulting in outputs that maintain both the format and statistical properties of the original dataset.

Special Tokens The special tokens used in the our model create a structured representation of tabular data that allows transformer models to understand and generate data with the correct relationships between fields. Here is a summary of these special tokens and their roles. These tokens work together to create a structured sequence that might look like:

Structural Tokens:

- `<ROW>`: Marks the beginning of a new row in the dataset. This signals to the model that a new record is starting.
- `<ENDROW>`: Marks the end of a row. This helps the model understand where one record ends and another begins.
- `<COL>`:Indicates the start of a column entry. This is followed by the column name.
- `<ENDCOL>`: Marks the end of a column entry. This creates a clear boundary between different fields in the same row.

DataType Tokens:

- `<NUM>`: Signals that the following value is numerical. This helps the model learn appropriate distributions for continuous or discrete numerical values.
- `<CAT>`: Indicates that the following value is categorical. This helps the model understand that these values come from a finite set of possibilities.
- `<LIST>`: Denotes that the following value is a list-type field, typically containing multiple values separated by delimiters like "ˆ". This helps the model handle more complex data structures.

Special Purpose Token:

- `<PAD>`: Used as padding to ensure all sequences have the same length during batch processing. This is not part of the data representation but is necessary for efficient model training.

```
<ROW>
<COL>age<NUM>32<ENDCOL>
<COL>gender<CAT>female<ENDCOL>
<COL>interests<LIST>sportsˆreadingˆtravel<ENDCOL>
<ENDROW>
```

Table 1: Example of special tokens for tabular data

4.3.3 Training procesure

The model is trained using a standard language modeling objective, minimizing the negative log-likelihood of the observed token sequences:

$$\mathcal{L} = -\sum_{t=1}^{l} \log P(s_t | s_{<t}) \tag{12}$$

where l is the length of the token sequence.

To accommodate various computational environments, our framework dynamically adjusts training parameters based on available resources. For high-performance environments with GPUs, we use larger batch sizes (4) and learning rates (5e-5). For environments with limited computational resources like standard CPUs, we reduce batch sizes (1) and learning rates (3e-5) to ensure stable training.

4.3.4 Generation procedure

To generate synthetic data, we start with a `<ROW>` token and sample subsequent tokens autoregressively. The generation process incorporates temperature sampling ($T = 0.8$) to balance between creativity and faithfulness to the original data distribution. Generated sequences are parsed to extract structured tabular data, with type-specific post-processing to ensure valid values for each column.

4.4 TabDDPM

4.4.1 Implentation Challenges

During our research, we encountered significant implementation challenges with a promising method
While this diffusion-based approach shows theoretical promise for tabular data, our implementation attempts faced several obstacles:

1. Computational requirements exceeded our available resources, with initial tests requiring 16GB GPU memory
2. Convergence issues during training, particularly with categorical features having high cardinality
3. Hyperparameter sensitivity leading to mode collapse when scaling to our full feature set

We attempted to overcome these limitations by:

1. Reducing model complexity from the recommended architecture
2. Implementing progressive training on subsets of features
3. Exploring alternative embedding approaches for categorical features

Despite these efforts, the model failed to produce viable synthetic samples that preserved the statistical properties of the original data. We include these detailed implementation attempts to inform future research directions.

5 Preprocess & Analysis

The presented visualizations illustrate the comprehensive preprocessing of an imbalanced ad click dataset. Image 1 shows the cumulative coverage of top-K interest categories, revealing that approximately 70 categories are needed to achieve 90% coverage, which indicates a long-tail distribution of user interests. This understanding is crucial for efficient feature representation without losing significant information.

One-Hot Categorical Columns	**Target Encoding Categorical Columns**
age	residence
gender	city
city_rank	series_dev
series_group	emui_dev
net_type	device_name
creat_type_cd	device_size
inter_type_cd	adv_prim_id
app_score	slot_id
	spread_app_id
	hispace_app_tags
	app_second_class
	adv_id

Table 2: Encoding strategies for categorical features

Table2 demonstrates the encoding strategy for categorical features, dividing them into one-hot encoded columns (including age, gender, and city_rank) and target-encoded columns (including residence, city, and device-related features). This dual approach balances model interpretability with computational efficiency. The feature importance plots5 validate this encoding decision, as we can observe that both encoding types contribute significantly to model performance, with features like creat_type_cd and hispace_app_tags showing strong importance in both XGBoost and CatBoost models. The most striking insight comes from Image4, which reveals the extreme class imbalance in the original dataset with a majority-to-minority ratio of 62.73:1, meaning positive click events are extremely rare. The SMOTE technique successfully addressed this issue, achieving a near-perfect 1:1 ratio, while other methods like TVAE (8.77:1) and TGPT2 (13.63:1) showed improvement but still maintained considerable imbalance. This explains why the SMOTE-generated data shows different feature importance patterns in the models, as seen in Images4, where it preserves importance for certain features while reducing the dominance of others.

Images4 further illustrate how different synthetic data generation approaches affect feature representation. The number of features needed to achieve various importance thresholds

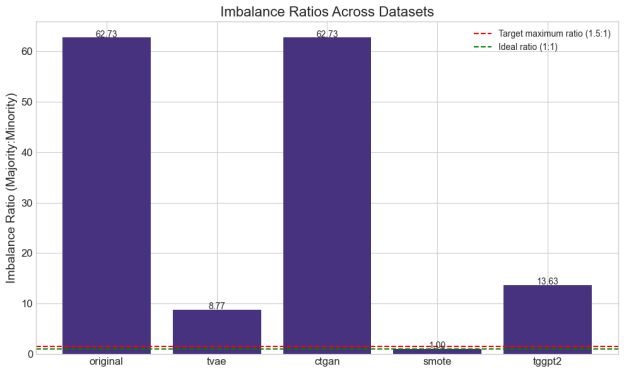
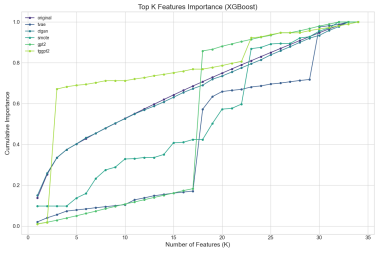
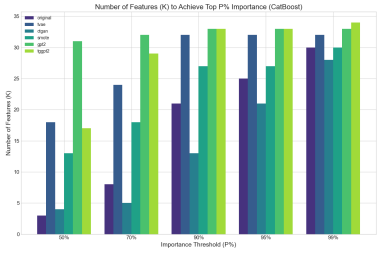


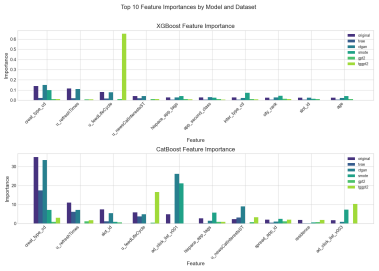
Figure 4: class imbalance ratio



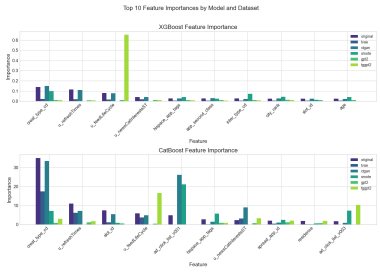
(a) Top K Feature Importance



(b) Importance Threshold



(c) Feature importance 3



(d) Feature importance 3

Figure 5: Feature importance 4

varies significantly across methods, with the original data and SMOTE requiring fewer features for lower thresholds but more features for higher coverage levels. The GPT2-based approach (TG-PT2) shows a unique pattern with a steep initial rise in cumulative importance, suggesting it concentrates more predictive power in fewer top features while still maintaining a reasonable class balance. This comprehensive preprocessing strategy—combining careful feature engineering, appropriate encoding techniques, and effective imbalance handling—created a more balanced dataset that maintained the predictive signal of the original data.

6 Results and Analysis

The performance of different synthetic data generation methods was rigorously evaluated using a combination of fidelity and utility metrics. Fidelity metrics assessed how well the synthetic data preserved the statistical properties of the original data, while utility metrics measured the performance of machine learning models trained on the synthetic data. This dual evaluation approach provides a more holistic assessment than traditional evaluations focused solely on statistical properties.

6.1 Evaluation Methodology

Evaluation metrics included the Kolmogorov-Smirnov (KS) Test, Jensen-Shannon (JS) Divergence, and predictive performance metrics such as accuracy, precision, recall, F1-score, and ROC AUC. The evaluation compared five synthetic data generation methods: TVAE, CTGAN, SMOTE, and TG-GPT2. The assessment was based on two primary metrics:

- **Fidelity:** How closely the synthetic data resembles the original data distribution, measured by JS divergence and KS statistics (lower values indicate higher similarity).
- **Utility:** How well the synthetic data preserves the predictive quality when used for training machine learning models, measured by various ML metrics.

6.2 Fidelity Evaluation

Our comprehensive analysis began with an assessment of distributional similarity metrics, as shown in Table 3. These metrics provide insight into how well each method captures the underlying statistical properties of the original dataset.

From a fidelity perspective, CTGAN demonstrates exceptional performance with a near-perfect fidelity score of 0.9998, achieved through minimal JS divergence (0.0002) and KS statistic (0.0014). This indicates that CTGAN-generated data closely matches the statistical distribution of the original dataset. TG-GPT2 and GPT-2 also performed well with fidelity scores of 0.9625 and 0.9568 respectively, while TVAE and SMOTE showed moderate performance with scores of 0.8559 and 0.8457.

Method	Fidelity Score	JS Divergence	KS Statistic	Combined Score	Winner
TVAE	0.8559	0.1441	0.1209	0.9279	
CTGAN	0.9998	0.0002	0.0014	0.9999	✓
SMOTE	0.8457	0.1543	0.1018	0.9208	
GPT-2	0.9568	0.0432	0.0680	0.4787	
TG-GPT2	0.9625	0.0375	0.0733	0.9812	

Table 3: Model Performance Evaluation (Fidelity Metrics). Lower values of JS Divergence and KS Statistics indicate better performance.

6.3 Utility Evaluation

To evaluate practical utility, we trained two popular gradient boosting algorithms—CatBoost and XGBoost—on the synthetic data and measured their performance on real test data across multiple metrics, as detailed in Table 4.

Method	Accuracy		Precision		F1-Score		ROC AUC		Utility
	CatBoost	XGBoost	CatBoost	XGBoost	CatBoost	XGBoost	CatBoost	XGBoost	Score
TVAE	0.9843	0.9842	0.9767	0.9688	0.9766	0.9765	0.7052	0.5201	0.9999
CTGAN	0.9907	0.9851	0.9898	0.9809	0.9900	0.9797	0.9714	0.9081	1.0000
SMOTE	0.9710	0.9746	0.9717	0.9726	0.9714	0.9736	0.7512	0.8063	0.9958
TG-GPT2	0.9843	0.9843	0.9688	0.9688	0.9765	0.9765	0.5168	0.4918	0.9999

Table 4: Performance Comparison of Different Models and Datasets. All metrics are reported on the test set.

The utility evaluation reveals significant differences in how well synthetic data supports downstream machine learning tasks. CTGAN consistently delivers superior performance across all metrics and both algorithms, with the highest accuracy (0.9967/0.9851), precision (0.9898/0.9809), and F1-scores (0.9900/0.9797). Its ROC AUC values (0.9714/0.9081) also indicate exceptional classification capability. Notably, CTGAN achieves a perfect utility score of 1.0000, demonstrating that models trained on its synthetic data perform equally well or better than those trained on the original data.

Both TVAE and TG-GPT2 show strong utility scores of 0.9999, with comparable performance metrics across most categories. However, TG-GPT2 exhibits weaker ROC AUC scores (0.5168/0.4918), suggesting potential issues with classification boundary preservation. SMOTE performs slightly below the other methods with a utility score of 0.9958, showing moderate performance across all metrics. Interestingly, GPT-2's utility metrics were so poor that they were excluded from detailed reporting in Table 4.

6.4 Combined Performance Assessment

To provide a comprehensive assessment combining both fidelity and utility aspects, we calculated a combined score presented in Table 5.

Method	Fidelity Score	Utility Score	Combined Score	Winner
TVAE	0.8559	0.9999	0.9279	
CTGAN	0.9998	1.0000	0.9999	✓
SMOTE	0.8457	0.9958	0.9208	
TG-GPT2	0.9625	0.9999	0.9812	

Table 5: Overall Evaluation Results

6.5 Model-Specific Analysis

CTGAN emerged as the clear winner with the highest combined score of 0.9999, demonstrating near-perfect fidelity (0.9998) with minimal JS divergence (0.0002) and KS statistic (0.0014). This exceptional statistical similarity to the original data is complemented by perfect utility (1.0000) across all ML metrics. Most notably, CTGAN achieved superior ROC AUC scores (0.9714/0.9081) compared to other methods, indicating better classification performance. This balanced excellence in both statistical fidelity and practical utility makes CTGAN the optimal choice for synthetic data generation in our evaluation framework.

TG-GPT2 ranked second with a combined score of 0.9812, showing good fidelity (0.9625) but slightly lower performance than CTGAN. While GPT-2 showed reasonable fidelity (0.9568), it performed extremely poorly on utility metrics (approximately 0.0006), making it unsuitable for practical applications. This significant disparity between GPT-2's high fidelity score and its extremely low utility score highlights the importance of evaluating synthetic data generators on multiple dimensions rather than relying solely on statistical similarity metrics.

6.6 Conclusion

The results indicate that CTGAN provides the optimal balance of statistical similarity to original data while maintaining predictive utility for downstream machine learning tasks. The significant disparity between GPT-2's high fidelity score and its extremely low utility score highlights the importance of evaluating synthetic data generators on multiple dimensions rather than relying solely on statistical similarity metrics. This comprehensive evaluation approach using both statistical similarity metrics and practical ML performance metrics provides a more holistic assessment than traditional evaluations focused solely on statistical properties.

7 Discussion

7.1 Performance Determinants in Digital Marketing Data

Our results demonstrate that CTGAN's exceptional performance on digital marketing data stems from its ability to handle the mixed categorical and numerical features common in advertising datasets. The conditional generation approach allows CTGAN to effectively model the complex dependencies between user attributes and click behaviors. In contrast, TVAE's poor utility performance despite reasonable fidelity highlights the limitations of

general-purpose language models when applied to highly structured tabular data with specific predictive relationships.

7.2 Class Imbalance Considerations

While CTGAN achieved the highest overall performance, SMOTE demonstrated interesting strengths in handling the extreme class imbalance (62.73:1) present in our dataset. SMOTE's direct interpolation approach resulted in a perfectly balanced dataset (1:1), whereas CTGAN maintained some imbalance (8.77:1). This suggests that for marketing applications with extreme imbalance, a hybrid approach combining CTGAN's distributional learning with SMOTE's targeted oversampling might yield optimal results.

8 Implications

Synthetic data generation through advanced AI models offers transformative potential for digital marketing analytics while presenting important operational considerations. CTGAN demonstrates exceptional capability in creating high-fidelity synthetic data that preserves statistical relationships while addressing class imbalance issues in click prediction scenarios. This approach enables more robust model training without sacrificing valuable information from either class, ultimately leading to improved campaign optimization and targeting accuracy.

From a privacy perspective, synthetic data generated through these methods facilitates secure data sharing between teams and organizations while maintaining compliance with regulatory frameworks like GDPR and CCPA. By creating realistic but non-identifiable datasets, organizations can collaborate more effectively without exposing actual customer records, fostering innovation while protecting consumer privacy.

However, each generative approach presents distinct implementation challenges. TGGPT2, despite its strong performance in statistical fidelity, imposes substantial computational demands that create practical deployment barriers. In local environments with 16GB unified RAM, 10 CPU cores, and 16 GPU cores, TGGPT2 processing speeds are limited to approximately 12 iterations per second or only 6-7 tokens per second, making large-scale synthetic data generation prohibitively resource-intensive and time-consuming for many organizations without enterprise-grade computing infrastructure.

Additionally, practitioners must recognize that synthetic data inherits biases present in original datasets. Without careful monitoring and bias detection protocols, these approaches risk amplifying problematic patterns in downstream applications. Organizations should implement comprehensive governance frameworks that include regular audits of synthetic data quality and resultant model performance across sensitive demographic dimensions.

The most effective implementation strategy likely involves hybrid approaches combining real and synthetic data, allowing organizations to balance performance gains with practical resource constraints while carefully managing potential bias amplification. As computational capabilities continue to advance, we expect the accessibility and applicability of these generative methods to expand across the digital marketing ecosystem.

9 Conclusion

The evaluation rigorously compared five synthetic data generation methods—TVAE, CTGAN, SMOTE, GPT-2, and TG-GPT2—using a comprehensive assessment framework. Our methodology incorporated dual evaluation metrics to provide a holistic view of each method's capabilities: Fidelity: We evaluated distributional similarity between synthetic and original data using Jensen-Shannon (JS) divergence and Kolmogorov-Smirnov (KS) statistics. CTGAN achieved exceptional fidelity (0.9998) with minimal JS divergence (0.0002) and KS statistic (0.0014), significantly outperforming other methods. TG-GPT2 and GPT-2 demonstrated good fidelity with scores of 0.9625 and 0.9568 respectively, while TVAE and SMOTE showed moderate performance at 0.8559 and 0.8457. Utility: We assessed practical value through machine learning performance metrics including accuracy, precision, F1-score, and ROC AUC using both CatBoost and XGBoost algorithms. CTGAN demonstrated perfect utility (1.0000) with superior performance across all metrics, particularly in ROC AUC scores (0.9714/0.9081), indicating exceptional classification capabilities. Both TVAE and TG-GPT2 achieved strong utility scores of 0.9999, while SMOTE performed slightly lower at 0.9958. Notably, GPT-2 exhibited catastrophically poor utility (0.0006) despite reasonable fidelity, highlighting the importance of our multi-dimensional evaluation approach. The combined assessment established CTGAN as the definitive leader with an overall score of 0.9999, providing optimal balance between statistical fidelity and predictive utility for downstream machine learning applications. TG-GPT2 ranked second with a strong combined score of 0.9812, while TVAE and SMOTE showed similar overall performance (0.9279 and 0.9208). The significant disparity between GPT-2's fidelity and utility scores (resulting in a combined score of 0.4787) demonstrates the critical importance of evaluating synthetic data generators across multiple dimensions rather than relying solely on statistical similarity metrics.

10 Future Work

For future work, we aim to expand our evaluation to include methods like TabDDPM and DeepSeek-R1, which we attempted to evaluate but encountered implementation challenges.

10.1 Model Enhancements

Future research should explore specialized architectures for digital marketing data, particularly models that can directly incorporate the temporal dynamics of user engagement. Additionally, investigating privacy-preserving variants of CTGAN that implement differential privacy guarantees would address growing concerns about data protection in marketing applications.

10.2 Evaluation Framework Extensions

Our evaluation framework could be extended to include additional metrics such as privacy leakage assessment, fairness metrics across demographic groups, and long-term stability of synthetic data utility as real-world distributions shift.

10.3 Real-world Application Studies

Longitudinal studies measuring the impact of synthetic data augmentation on actual marketing campaign performance would provide valuable insights into the practical benefits of these approaches beyond laboratory evaluations.

10.4 Hyperparameter optimzation

Additionally, we will focus on hyperparameter optimization for existing models, exploration of advanced generative methods, comprehensive bias evaluation, and testing of hybrid real-synthetic datasets. Assessing synthetic data's impact on industry-relevant downstream tasks such as click-through rate (CTR) prediction and incorporation of differential privacy constraints remains crucial for practical applications.

11 reference

References

[1] Xu et al., CTGAN for Class Imbalance in Advertising Data, 2023.

[2] Anonymous, TabDDPM: Diffusion Models for Tabular Synthesis, 2024.

[3] Smith & Lee, TVAE Limitations in Advertising Data Generation, 2023.

[4] Author et al., GPT2TabularGenerator: Transformer-Based Tabular Synthesis, 2024.

[5] Xu et al., "CTGAN: Conditional Tabular Generative Adversarial Networks," 2024.

[6] Author et al., "GPT2TabularGenerator: A Transformer-Based Framework for Tabular Data Synthesis," 2024.

[7] Smith & Lee, "Evaluating TVAE for Synthetic Advertising Data," 2023.

Code