mathematics

MDPI

Article

On the Quality of Synthetic Generated Tabular Data

Erica Espinosa [1] and Alvaro Figueira [2,3,*]

[1] Department of Mathematics Engineering, Politecnico di Milano, 20133 Milan, Italy
[2] Faculty of Sciences, University of Porto, 4169-007 Porto, Portugal
[3] INESCTEC, 4200-465 Porto, Portugal
* Correspondence: arfiguei@fc.up.pt

Abstract: Class imbalance is a common issue while developing classification models. In order to tackle this problem, synthetic data have recently been developed to enhance the minority class. These artificially generated samples aim to bolster the representation of the minority class. However, evaluating the suitability of such generated data is crucial to ensure their alignment with the original data distribution. Utility measures come into play here to quantify how similar the distribution of the generated data is to the original one. For tabular data, there are various evaluation methods that assess different characteristics of the generated data. In this study, we collected utility measures and categorized them based on the type of analysis they performed. We then applied these measures to synthetic data generated from two well-known datasets, Adults Income, and Liar+. We also used five well-known generative models, Borderline SMOTE, DataSynthesizer, CTGAN, CopulaGAN, and REaLTabFormer, to generate the synthetic data and evaluated its quality using the utility measures. The measurements have proven to be informative, indicating that if one synthetic dataset is superior to another in terms of utility measures, it will be more effective as an augmentation for the minority class when performing classification tasks.

Keywords: utility measures; synthetic data; class imbalance; tabular data

MSC: 68T99

Citation: Espinosa, E.; Figueira, A. On the Quality of Synthetic Generated Tabular Data. *Mathematics* **2023**, *11*, 3278. https://doi.org/10.3390/math11153278

Academic Editors: Jose Antonio Sáez Muñoz and José Luis Romero Béjar

Received: 18 June 2023
Revised: 21 July 2023
Accepted: 24 July 2023
Published: 26 July 2023

1. Introduction

Predictions made through classification models are nowadays used in many fields, but a common challenge that arises is class imbalance. This is a well-known problem in machine learning: it occurs when one or more classes in a dataset are significantly underrepresented compared to other classes. There are several problems associated with a class imbalance in machine learning as discussed in [1,2]. The algorithms trained on imbalanced data tend to be biased towards the majority class, which leads to poor performance in predicting the minority class. For example, in the field of medicine, rare diseases often lack sufficient data for analysis causing problems in decision-making [3], as well as identifying security bugs from a bug repository [4] or traffic accidents [5]. Fraud detection is also hindered by imbalanced datasets [6,7]. Additionally, fake news detection represents a common scenario of class imbalance, necessitating addressing this issue prior to algorithms development [8,9]. In recent years, to address this issue, synthetic data augmentation has been developed to increase the representation of the minority class. Some examples of this approach include the generation of synthetic images [10,11], time series [12], and tabular data [13,14]. The aim is to increase the amount of data available for the underrepresented class, thereby enhancing the performance of machine learning models in accurately predicting this class [15–17]. One way to upsample the minority class is through the generation of synthetic data, that is, data not from actual sampling, but created by generative models that attempt to emulate the same distribution as real data.

There are several ways to generate synthetic data. One common method to generate new samples is the SMOTE algorithm [18] (and variations of it), which generates new data by interpolating the original available ones. Other very common generative models are generative adversarial networks (GANs) [19], and variational autoencoders (VAEs) [20]. However, to ensure that these synthetic data are useful and do not only add noise to our real dataset, it is important to verify and evaluate whether they are representative of the real sample. Therefore, we need objective tools to compare the synthetic data to the real data and then evaluate the differences.

In this paper, we aim to explore the utility measures for assessing the quality of synthetic data generated from real datasets, particularly in the context of tabular data. Our research aims to investigate and analyze the utility measures that can effectively quantify the difference between real and synthetic tabular data. By examining the applicability of these measures in the context of real-world datasets, we aim to provide valuable insights into the assessment of synthetic tabular data quality. Finally, a crucial aspect of our research is to evaluate the effectiveness of data augmentation using synthetic data specifically in the context of tabular data classification tasks. By incorporating synthetic data into the training process, we aim to examine its impact on improving classification performance. Through this evaluation, we not only assess the performance of the classification models but also gain insights into the generative methods employed. The ability of these methods to accurately learn and capture the underlying distribution of the data is therefore intrinsically evaluated. We consider that this evaluation provides valuable information regarding the suitability and effectiveness of the generative methods in generating synthetic data that closely aligns with the characteristics and patterns of the original dataset. By thoroughly assessing the classification performance and considering the impact of data augmentation, we can shed light on the potential of synthetic data for improving classification accuracy and expanding the capabilities of machine learning models in handling tabular data. Moreover, the evaluation of different generative methods contributes to advancing the field of generative modeling, providing insights into the strengths and limitations of these methods in learning complex data distributions.

In Section 2, we present the statistical utility measures categorized based on their ability to capture different types of information. Section 3 examines how we can determine the usefulness of the generated data for our classification task. The datasets used in this study are described in Section 4. Section 5 provides a list and description of the generative methods employed to create the synthetic data used for augmentation. Lastly, Section 6 presents the results obtained in our analysis.

2. Statistical Utility Measures

To ensure that synthetic data accurately represents the characteristics of the real data, it is essential to evaluate the similarity between their distributions. Utility measures, also known as evaluation metrics, are used to assess the performance or effectiveness of a system, model, algorithm, or any other process. This type of evaluation comes in handy for our purpose: in order to assess the goodness or truthfulness of synthetic data we want to measure their similarity to real data in terms of distribution. Utility measures are divided into three categories based on how thoroughly they investigate the distributions. In *univariate* measures, the focus is on preserving the individual distributions of each column in the original data. This is achieved by comparing the similarity between the synthetic data and the original data column-by-column. The second category, known as *bivariate* extends the measure to consider the correlation between the columns being studied. Hence, it is a pairwise study. Finally, the third category, *multivariate*, examines the joint distribution of all the columns together, translating into a comparison made between two complete datasets.

In the following sections, we revisit some of these techniques. We use the following notation to refer to both the real and synthetic datasets: let $X = (X_1, \ldots, X_d)$ be a multivariate random variable with distribution F, where each component $X_{i=1,\ldots,d}$ represents one of

the d columns of the real dataset. Similarly, let $Y = (Y_1, \ldots, Y_d)$ be a random variable with distribution G, representing the synthetic dataset. We denote the l-th row of the real dataset as x_l, which represents a realization of the random variable X. The entire real dataset can be viewed as a collection of such realizations and is denoted as $\mathsf{X} = \{\mathsf{x}_l\}_l$. The same applies to the synthetic dataset, which is indicated as $\mathsf{Y} = \{\mathsf{y}_m\}_m$.

2.1. Univariate Measures

To assess the similarity between the distributions of the synthetic and real data, it is necessary to examine their univariate distributions. Specifically, we need to measure the deviation in distribution between each pair of corresponding variables X_i and Y_i for $i = 1, \ldots, N$. This distance reflects the differences in distribution between the same columns of the two datasets. These measures are the easiest to measure and interpret, but nonetheless fundamental in the analysis of synthetic data because if there is already a large difference between the actual and synthetic data at this level, the similarity is unlikely to increase by considering more variables at a time. Univariate measures yield a distinct value for each variable, which can be further analyzed through visualization techniques such as boxplots.

One of the most widely used measures for assessing the univariate distance between two distributions is the Hellinger distance [21]. It is defined as follows: let P and Q denote two probability measures on a measure space $\mathcal{X}$, the Hellinger distance between P and Q is

$$H(P,Q) \quad = \quad \frac{1}{\sqrt{2}}\sqrt{\int_{\mathcal{X}}\left(\sqrt{P(dx)} - \sqrt{Q(dx)}\right)^2}. \tag{1}$$

Since we have only the realizations of the probability measures F and G, for each variable i we will adapt the distance to the discrete case: $F_i = (f_1, \ldots, f_k)$ and $G_i = (g_1, \ldots, g_k)$,

$$H(F_i,G_i) \quad = \quad \frac{1}{\sqrt{2}}\sqrt{\sum_{j=1}^{k}\left(\sqrt{f_j} - \sqrt{g_j}\right)^2}, \tag{2}$$

where F_i and G_i are the marginal distribution of the i-th component of X and Y, respectively, and f_j and g_j are the proportion of counts of instances for the j-th bin of the interval of values for each variable i.

Another widely used measure is the Kullback–Lieber divergence [22]. Specifically, the Kullback–Leibler divergence of Q from P denoted $D_{\mathrm{KL}}(P\|Q)$, is the measure of information lost when Q is used to approximate P. In our discrete case, it is defined as:

$$D_{\mathrm{KL}}(F_i\|G_i) \quad = \quad \sum_{j=1}^{k} f_j \log \frac{f_j}{g_j}. \tag{3}$$

When comparing two probability distributions using the Kullback–Liebler divergence, it is important to note that the supports of the distributions must match, otherwise the divergence may be infinite due to the presence of zero probabilities in the G_i distribution. When the supports do not match, it means that there are some values that have non-zero probability in one distribution but zero probability in the other.

To address this issue, the Jensen–Shannon divergence [23] was proposed, which is a symmetrized version of the Kullback–Liebler divergence that avoids this problem by smoothing out the differences between the supports of the two distributions:

$$D_{\mathrm{JS}}(F_i||G_i) \quad = \quad \sqrt{\frac{1}{2}\left[D_{\mathrm{KL}}\left(F_i||M_i\right) + D_{\mathrm{KL}}\left(G_i||M_i\right)\right]} \tag{4}$$

with

$$M_i \quad = \quad \frac{F_i + G_i}{2}.$$

Specifically, it is calculated as the square root of the average of the Kullback–Liebler divergence between each distribution and their average. This makes it a useful measure for comparing probability distributions, even when their supports do not match. It is always non-negative and ranges from 0 (when the distributions are identical) to 1 (when they have no common support).

2.2. Bivariate Measures

To ensure that the synthetic data accurately represent the relationships among the variables in the original dataset, we need to use bivariate utility measures. While univariate measures can be useful for evaluating the overall distribution of each variable separately, they do not capture the complex relationships that may exist between multiple variables. By using bivariate measures, we can verify that the same underlying relationships between variables are maintained in the synthetic data as well. To assess this, the pairwise correlation difference (PCD) [24] measures how much the correlation between real and synthetic data differs. The PCD is defined as:

$$PCD(\mathsf{X}, \mathsf{Y}) \quad = \quad \|Corr(\mathsf{X}) - Corr(\mathsf{Y})\|_F, \tag{5}$$

where X, Y are the real and synthetic datasets, respectively. $\|\cdot\|_F$ is the Frobenius norm [25], which for an $m \times n$ matrix A is defined as:

$$\|\mathsf{A}\|_F \quad = \quad \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2},$$

where a_{ij} denotes the (i, j)-th entry of the matrix A. The choice of this norm as a measure for the PCD is motivated by its intuitive interpretation as a matrix size measure and its similarity to the L^2 norm for vectors, as it shares similar properties with the latter.

2.3. Multivariate Measures

While univariate and bivariate measures can provide valuable insights into individual variables and their pairwise relationships, they fail to fully capture the intricate multivariate relationships that exist within both the original and synthetic datasets. To ensure that the synthetic data accurately represents the higher-order dependencies and interactions among variables in the original dataset, we need to use multivariate measures. Multivariate measures are designed to assess the joint distribution of three or more variables, providing a deeper understanding of the underlying patterns and structures that may be present in the data. In particular, they can be used to quantify the difference between the multivariate distributions of two datasets, which in our case is $|F - G|$.

One way to do this is through the Kolmogorov–Smirnov test statistics that compare the empirical cumulative distribution functions of the real and synthetic cases:

$$D_m \quad = \quad \max_n |\mathcal{F}_X(\boldsymbol{z_n}) - \mathcal{F}_Y(\boldsymbol{z_n})| \tag{6}$$

$$D_s \quad = \quad \frac{1}{N}\sum_n [\mathcal{F}_X(\boldsymbol{z_n}) - \mathcal{F}_Y(\boldsymbol{z_n})]^2 \tag{7}$$

where $\mathcal{F}_X$ and $\mathcal{F}_Y$ are, respectively, the empirical cumulative density functions for X and Y, $\boldsymbol{z_n}$ is a d dimensional vector representing a possible instance for the two random variables, and N is their total number. The deviation of D_m and D_s from zero represents the distance between the two distributions, the former in terms of the maximum absolute difference capturing the largest deviation, the latter in terms of mean squared difference which quantifies the overall distance between the distributions. Unfortunately, this measure encounters two primary challenges. Firstly, the curse of the dimensionality problem undermines its accuracy when applied to high-dimensional data, as estimating multivariate distributions becomes increasingly complex. Secondly, the test's sensitivity to changes in

the tail of the cumulative distribution may hinder its ability to detect significant differences in other regions of the distribution [26].

The propensity score [27] evaluates the distance between the two datasets through a classification model. The original and synthetic datasets are assigned distinct labels to enable differentiation, subsequently combined, and provided as input to a classifier. The propensity score measures how well the classifier is able to distinguish the two types of data and it is defined as follows:

$$pMSE = \frac{1}{N}\sum_{n}(\hat{p}_n - 0.5)^2 \quad (8)$$

where p_n is the probability of belonging to the synthetic class assigned by the classifier to each instance. The less the classifier is able to distinguish between the two classes (thus returning p_n values of roughly 0.5) the closer the propensity score is to zero. In their study, Snoke et al. [27] proposed the standardized pMSE (St pMSE), which quantifies the difference between the expected and observed values in terms of the estimated null standard deviation. Higher values of this measure indicate a poorer fit between the data and the underlying synthesis, while values closer to zero indicate a better fit. Rescaling by the null statistic provides a more intuitive measure of its utility, particularly when applied to synthetic data. Notably, this rescaling renders the measure independent of sample size, allowing for easier comparisons across datasets.

Another useful criterion for evaluating the quality of synthetic data is the order of variable relevance, which can be assessed by constructing decision trees on both real and synthetic datasets and comparing the resulting variable orders. Different methods can be employed to measure the distance between the two sequences, such as focusing on the first three variables or considering the majority of the order.

3. Application-Specific Measures: Classification

When generating synthetic data to perform specific tasks, it is recommended to evaluate their quality by comparing the performance of real and synthetic data to complete these jobs. In this study, our objective is to examine the utility of synthetic data in enhancing the performance of a classification model when dealing with class imbalance. To achieve this, we augment the minority class by introducing synthetic data into the training set. Subsequently, we compare the model's performance on the test set with the performance of a model trained exclusively on the original data. The higher the classifier's performance, the more the generated data are similar to the minor class and hence more useful to the classifier.

4. The Datasets

In order to assess the effectiveness of generative models in creating realistic synthetic samples, we evaluate the performance of different utility measures on two real-world application datasets: the Adults Income dataset (http://archive.ics.uci.edu/dataset/2/adult, accessed on 17 March 2023) [28] and the Liar+ dataset (https://www.cs.ucsb.edu/~william/data/liar_dataset.zip, accessed on 17 March 2023) [29]. The Adults Income dataset was chosen for its widespread use in the literature [30–32] and its relevance to socioeconomic factors that influence income. The Liar+ dataset is a well-established dataset that finds extensive usage in the domains of natural language processing and machine learning [33–35]. We selected it due to its different nature with respect to the Adults dataset and because it represents a typical example of class imbalance in the real world. Moreover, following the preprocessing steps outlined in the forthcoming sections, the dataset will feature a substantial number of covariates, presenting a challenge in generating synthetic data.

The Adult Income dataset, also known as the Census Income dataset, is a widely used dataset in machine learning and data analysis. It contains demographic and employment-related information on 48,842 individuals from the 1994 US Census database. The dataset consists of 13 attributes (six continuous and seven categorical) including the individual's

age, education level, occupation, marital status, relationship, race, gender, and native country. The target variable of the dataset is the income bracket, divided into two classes: "$\leq$50 K" with 41,001 samples, and "$>$50 K" with 7841 cases, indicating whether an individual's income is below or above 50,000 USD per year. To reduce the complexity of the categorical variables and ensure the feasibility of the analysis, we transformed them into binary variables using the following criteria:

- For variables "race" and "native country" there was one class representing more than 85% of the cases, therefore we selected label 1 for the predominant class and 0 for all the others.
- We transformed the other categorical variables into binary variables by grouping cases that shared similar conceptual characteristics.

For example, if the original variable was "native country" and the most common category was "USA", then the new binary variable would have a value of 1 for "USA" and 0 for all other categories. The dataset was partitioned into training and test sets to enable the evaluation of classifiers based on application-specific measures. The data were randomly partitioned in a manner that allocated 80% of the samples to the training set and 20% to the test set while ensuring that the original balance between the classes was maintained. The training set consisted of 32,801 samples in Class 0 and 6273 samples in Class 1, while the test set comprised 8200 samples in Class 0 and 1568 samples in Class 1.

The Liar+ dataset is a dataset of statements made by politicians that have been labeled as "true", "mostly true", "half true", "barely true", "false", and "pants on fire". The dataset utilized in this study underwent the following preprocessing steps: only samples belonging to the "pants-fire", "false", "mostly true", and "true" classes were retained, with the first two classes merged and labeled as 1, and the latter two labeled as 0. To address the class imbalance observed in the real world, the class labeled as 1 was downsampled to a third of the size of the class labeled as 0, resulting in 4087 instances in class 0 and 1362 instances in Class 1. The remaining data processing followed the methodology outlined by Bruno Vaz in his thesis [36] which is the following: categorical variables such as "speaker", "job", "state", "party", and "subject$_i$" were subjected to target encoding, while the variables "statement", "context", and "justification", which contained essential information for predicting the target variable, underwent processing using the Doc2Vec algorithm. After preprocessing, the dataset consisted of 50 features and one target variable. Finally, the dataset was split into training and test sets with an 80/20% ratio, maintaining the proportional distribution of classes in each set. The resulting training set was composed of 3615 samples in Class 0 and 1216 samples in Class 1, while the test set consisted of 904 samples in Class 0 and 304 samples in Class 1.

5. Synthetic Data Generative Methods

We utilized several techniques for data synthesis, each serving a specific purpose. These included Borderline SMOTE [37], which is an important variation of the widely used upsampling technique SMOTE. We also employed DataSynthesizer [38] as one of the simplest of the models that attempt to sample from the distribution of real data. Additionally, CTGAN [39], a leading GAN model for tabular data, was utilized. We also explored CopulaGAN [40], as an improvement upon the CTGAN approach. Finally, we incorporated REaLTabFormer [41], which introduces a novel approach to tabular data synthesis by leveraging large language models and transformer architectures.

5.1. Description of the Generative Methods

Borderline SMOTE is a variant of the more traditional SMOTE (synthetic minority over-sampling technique) [18] algorithm that generates new data through the interpolation of existing data. It is characterized by interpolating only the minority class data that are close to the decision boundaries, allowing the algorithm to focus on examples that are more difficult to correctly classify. The approach is as follows: the nearest neighbors algorithm first identifies instances close to the decision boundaries, called borderline instances, and

then the synthetic samples are generated by randomly selecting a borderline instance and interpolating between it and one or more of its *k*-nearest neighbors, which are also minority class instances. The main advantage of Borderline SMOTE over the original SMOTE algorithm is that it can help to reduce the problem of overfitting, which can occur when synthetic samples are generated from all minority class instances, including those that are easy to classify.

DataSynthesizer is a system that generates synthetic datasets based on private input datasets. It consists of two main modules such as DataDescriber, and DataGenerator. DataDescriber infers attribute types and domains, supporting numeric, categorical, and datetime data. It calculates value frequency distributions for categorical attributes and equi-width histograms for numeric and datetime attributes. The module handles missing values and incorporates differential privacy by adding Laplace noise. DataGenerator takes the dataset description generated by DataDescriber and generates synthetic data by sampling from the specified distributions. It offers three modes: random mode, independent attribute mode, and correlated attribute mode. In the random mode, values are generated randomly based on attribute types. In the independent attribute mode, sampling is performed from bar charts or histograms using uniform sampling. Finally, in the correlated attribute mode, values are sampled in the appropriate order from a Bayesian network. In our study, we utilize the correlated attribute mode to incorporate intra-feature relationships into consideration. This probabilistic model is constructed using a Bayesian network that captures the correlation structure between attributes. Once the Bayesian network is obtained, it determines the order in which attribute values are sampled, resulting in the generation of synthetic data.

The CTGAN (conditional tabular generative adversarial network) model is a GAN-based approach designed to model the distribution of tabular data and generate synthetic rows that follow the same distribution. The popularity of this model in the world of synthetic tabular data stems from its ability to tackle issues related to non-Gaussian and multimodal distributions, as well as imbalanced discrete columns. Each continuous variable is remodeled using a variational Gaussian mixture (VGM) model, where the number of modes in the distribution is estimated. For each instance, the nearest mode is indicated using a one-hot vector, and a scale parameter is determined. This allows for flexible representation of continuous values within their respective modes. The CTGAN model incorporates a conditional generator and training-by-sampling to address imbalanced discrete columns. The conditional generator enables the generation of synthetic rows conditioned on a specific discrete column value. The generator is penalized during training to produce an exact copy of the given condition, ensuring that the generated rows preserve the specified condition. Training-by-sampling ensures that all categories from discrete attributes are sampled evenly (but not necessarily uniformly) during the training process, facilitating the exploration of all possible discrete values. The network structure of CTGAN consists of fully connected layers in both the generator and discriminator. Overall, CTGAN provides a solution for modeling tabular data distributions, handling complex distributions, imbalanced discrete columns, and generating synthetic rows that resemble the original data distribution.

The synthetic data vault CopulaGAN model can be seen as an elaboration of the CTGAN model. It employs a two-stage approach to generate synthetic data while preserving the statistical properties of the original dataset. In the first stage, known as statistical learning, the synthesizer focuses on learning the marginal distributions of the columns in the real dataset. This involves understanding the shape and characteristics of each individual column's distribution. For instance, it might identify a column as having a Beta distribution with parameters $\alpha = 2$ and $\beta = 5$. The synthesizer normalizes the values to a Gaussian distribution using the learned distributions. In the second stage, the synthesizer employs CTGAN to train the normalized data. By combining statistical learning with GAN-based modeling, the SDV CopulaGAN synthesizer generates synthetic data that not only captures

the individual column characteristics but also maintains the complex dependencies and structure present in the original dataset.

In conclusion, REaLTabFormer introduces an innovative approach to generating realistic tabular data as it is a transformer-based framework designed for generating non-relational and relational tabular data. In this study, we focus only on the generation of non-relational data. Therefore, it treats each observation as a sequence and learns the conditional distribution of columnar values in each row. This allows it to generate the next values in the sequence, effectively creating realistic non-relational data. The underlying architecture used for this purpose is GPT-2, a transformer-based autoregressive model. GPT-2 is known for its ability to capture the conditional distribution of sequential data effectively. To encode the tabular data efficiently, a fixed-set vocabulary is adopted for each column. This means that a predetermined set of possible values is defined for each column, and the model uses these values during the generation process. To address overfitting and improve the quality of the generated samples, REaLTabFormer incorporates target masking during training. Target masking involves replacing the target or label tokens with a special mask token. Hence, it forces the model to learn the masks instead of the actual values, encouraging it to generalize and generate more diverse samples. During the generation process, the model fills the masked values with probabilistically valid tokens. By leveraging the learned distribution, the model selects appropriate values from the predefined vocabulary.

This approach ensures that the generated non-relational data adhere to the patterns and distributions observed in the training data. In summary, the REaLTabFormer model uses an autoregressive approach and the GPT-2 architecture to generate realistic non-relational tabular data. By treating each observation as a sequence and learning the conditional distribution of columnar values, the model can accurately generate the next values in the sequence. The use of a fixed-set vocabulary, target masking during training, and probabilistic sampling further enhance the quality and diversity of the generated non-relational data.

5.2. Experiments on the Generative Methods

We recall that the goal of this process is to create new data using these models so that it is possible to add synthetic data to the minority class and improve the performance of the classifiers. To avoid overfitting and better evaluate the performance of our classification models, we trained the oversampling algorithms only on samples belonging the Class 1 of the training set. This allowed us to more accurately assess the generalization performance of our models and determine their ability to handle imbalanced data.

The Borderline SMOTE algorithm automatically recognizes the class with fewer samples and generates a total of synthetic samples so that the cardinality in both classes is the same. The first target class in the Adults Income dataset has 32,800 samples, while the minor class has 6273 samples; thus, the method generated 26,527 synthetic data. Instead, the difference between the two classes in the Liar+ dataset was 2399; therefore, these many samples were generated. The other models generated a different amount of samples each time, resulting in distinct samples in each run. We created a series of 1000, 5000, 10,000, and 20,000 samples for the Adults Income dataset and 500, 1000, 1500, and 2000 samples for the Liar+ dataset. The intrinsic motivation is to assess the effect of sample size on utility measures while ensuring that the sample size does exceed the number of samples in the majority class.

6. Results and Discussion

The analysis is categorized based on the used dataset. Initially, we examined the Adults Income dataset, followed by the Liar+ dataset. We assess the synthetic data using utility measures and subsequently evaluate its classification performance.

The synthesis algorithms are outlined in the following study on the Adults Income dataset: from this point forward when we write SMOTE, we mean Borderline SMOTE and

DS_{1k}, DS_{5k}, DS_{10k}, DS_{20k} stand for, respectively, the datasets made by 1000, 5000, 10,000, and 20,000 samples from the DataSynthesizer model. The same follows for the datasets generated by CTGAN, CopulaGAN, and REaLTabFormer, the last labeled as RTF. We begin by analyzing the quality of the generated data with the univariate measures, that is, we want to verify the similarity, column by column, of the generated dataset compared to the real dataset. Because univariate measures return a value for each component of the random variable that represents our dataset, i.e., our columns, we opted to represent them using boxplots. The more the values are grouped to values close to zero, the better the similarity between the real dataset and the generated one. The boxplots of the Jensen–Shannon Measures for the Adults Income dataset are shown in Figure 1.

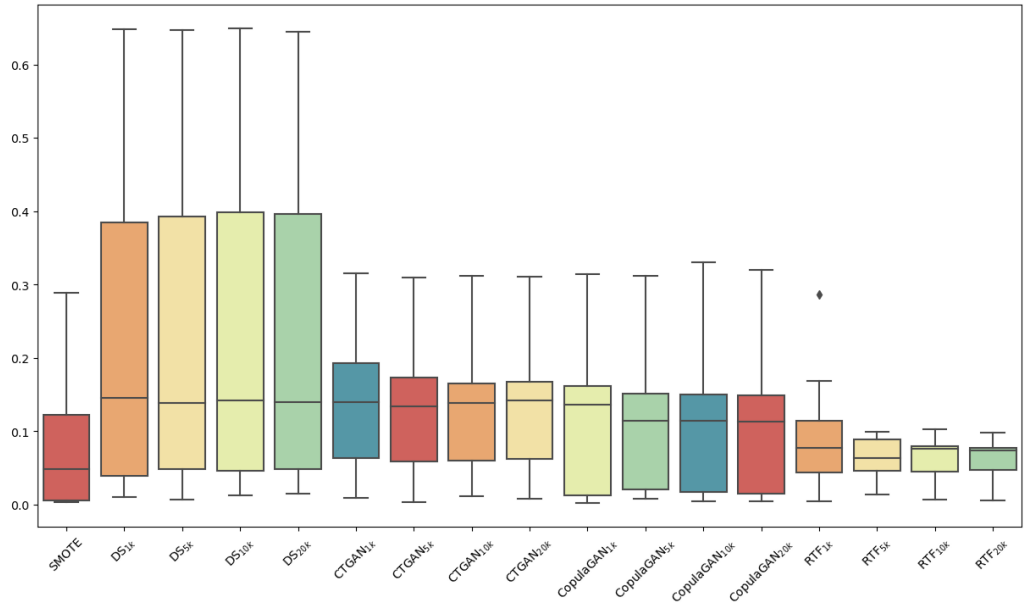

Figure 1. Boxplots of the Jensen–Shannon measures grouped by each synthesis method for the Adults Income dataset. On the *y*-axis there are the values of the distance measured on each column of the data frame.

The scores show little sensitivity to the sample size, suggesting that a relatively small sample of 5000 points already provides a robust representation of the variable distribution for generative models. Notably, among the models examined, REaLTabFormer shows a marginal improvement in performance. Among the methods compared, DataSynthesizer had higher median and variability scores, demonstrating a limited ability to capture the variables' distribution. REaLTabFormer and Borderline SMOTE achieved better results as the former had lower variability and the latter had lower median scores. CopulaGAN had lower median scores compared to the CTGAN but showed greater variability in scores. Table 1 presents the pairwise correlation difference (PCD) values in bivariate analysis, which evaluates the generative methods' capability to preserve the same interaction between the components as observed in the real dataset.

Table 1. Pairwise correlation difference in the Adults Income dataset for each synthesis method used.

	PCD		***PCD***
SMOTE	0.3387		
DS_{1k}	1.2528	$CopulaGAN_{1k}$	1.2812
DS_{5k}	1.1521	$CopulaGAN_{5k}$	1.1126
DS_{10k}	1.1113	$CopulaGAN_{10k}$	1.1325
DS_{20k}	1.1407	$CopulaGAN_{20k}$	1.1472
$CTGAN_{1k}$	1.3441	RTF_{1k}	0.3787
$CTGAN_{5k}$	1.3120	RTF_{5k}	0.2894
$CTGAN_{10k}$	1.3447	RTF_{10k}	0.2404
$CTGAN_{20k}$	1.3340	RTF_{20k}	0.2584

The value for the synthetic data reduces slightly as the sample size increases. This may be due to the fact that as datasets increase in size, the relationships among variables are more represented. Yet, in terms of maintaining interactions between variables, the Borderline SMOTE and REaLTabFormer methods appear to perform better.

To proceed to multivariate analysis, the propensity score was computed using a CART (otherwise known as decision tree) model in order to calculate the distinguishability between the original and synthetic datasets.

Table 2 shows the obtained values.

Table 2. Propensity score and standardized propensity score for the synthetic data generated from the Adults Income dataset.

	pMSE	***St pMSE***		***pMSE***	***St pMSE***
SMOTE	0.0433	Inf			
DS_{1k}	0.1186	351.4	$CopulaGAN_{1k}$	0.1102	597.1
DS_{5k}	0.2468	122.2	$CopulaGAN_{5k}$	0.2287	88.91
DS_{10k}	0.2369	3087.0	$CopulaGAN_{10k}$	0.2232	3524.1
DS_{20k}	0.1818	Inf	$CopulaGAN_{20k}$	0.1724	Inf
$CTGAN_{1k}$	0.2311	94.6	RTF_{1k}	0.0988	742.3
$CTGAN_{5k}$	0.2468	108.8	RTF_{5k}	0.2127	137.5
$CTGAN_{10k}$	0.2252	2203.0	RTF_{10k}	0.2080	5946.3
$CTGAN_{20k}$	0.1735	Inf	RTF_{20k}	0.1588	10,281.6

The closer the pMSE and Standardized pMSE are to 0, the better the indistinguishability between the two datasets under consideration. Because the null distribution of the pMSE is not theoretically derivable in this manner, it is estimated through resampling (details in [27]). After examining the scores of the models that were sampled with varying numbers of samples, it appears that pMSE follows a particular trend. In most cases, its value is the smallest when the sample size is low, increases to the maximum at 5000 samples, and then decreases again with 20,000 samples. In contrast, St pMSE exhibits an opposite trend to that of pMSE and even reaches infinite values with the largest sample size. In [27], the authors describe this indicator as unstable in the case of high sample sizes. Alternatively, this trend may be interpreted as follows: since pMSE employs a classifier to evaluate the quality of synthetic data, it is affected by class imbalance. With 6273 real samples, the point where the number of real and synthetic samples is most similar is when 5000 are generated by the models. Therefore, pMSE may be more significant in this case, as it scores lower for every model. A comparison of the performance of the 26,527 data generated by Borderline SMOTE with the other cases that generated 5000 samples is not meaningful in this context. Therefore, among the remaining models, the most suitable one for replicating the distribution of the real dataset is REaLTabFormer. And, since St pMSE is not affected by sample size, it can determine the optimal value in this case.

Finally, considering the application of the dataset for binary classification modeling, we evaluated and compared the performance of various algorithms—decision trees, logistic regression, random forest, and XGBoost. This comparison was made between the results from the datasets augmented with synthetic data and the baseline performances to observe any potential improvements or differences. The more the performances improve, the more the generated data help the minority class by increasing its representation. This means that the generated data comes from the same distribution as the minority class. Table 3 displays the F1 scores for Class 1 across all classifiers and synthesis methods computed on the test set. The classification models have been fitted on all the samples from Class 0 and on the real samples of Class 1, augmented with 20,000 synthetic instances to mitigate class imbalance.

Decision trees are found to be quite insensitive to the addition of new samples for Class 1, regardless of how they were generated as the F1 score improves only slightly with the augmentation. Logistic regression, on the other hand, which is a model that is very sensitive to class imbalance, shows significant performance improvement with class augmenting, especially if the data were generated using Borderline SMOTE, as the F1 score from a baseline value of 18.24 reached a value of 76.02. Figure 2 illustrates how the performance of the classifiers changes with varying numbers of added synthetic samples.

Table 3. F1 score in percentage on Class 1 for the classification on the Adults Income dataset on the test set in the case of Class 1 augmentation with 20,000 synthetic data.

	Decision Tree	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	40.14	18.24	35.29	34.57
Borderline SMOTE	41.31	**76.02**	**50.36**	**49.55**
DataSyntesizer	41.11	35.59	34.67	34.50
CTGAN	41.05	61.47	37.50	34.24
CopulaGAN	**41.79**	67.47	38.24	33.86
REaLTabFormer	40.56	66.20	42.69	39.16

As previously seen, the decision tree model does not appear to be sensitive to data augmentation while the other classification models show an increase in accuracy, precision, and F1 score as the number of samples increases. There is also a decrease in the recall, potentially indicating a reduction in model bias. Also here, logistic regression is shown to be very sensitive to data augmentation, and the performances increase as the number of samples added increases.

We computed the same measures for the Liar+ dataset. Nevertheless, in this situation, $DS_{0.5k}$, DS_{1k}, $DS_{1.5k}$, and DS_{2k} correspond to samples with a cardinality of 500, 1000, 1500, and 2000, respectively, and the same goes with the other generative models. Figure 3 depicts the boxplots of the univariate Jensen–Shannon measures. In this scenario, the Borderline SMOTE algorithm and the REaLTabFormer outperform the other models showing a median that is almost half of all the other cases and very low variability. Again, the DataSynthetsizer model turns out to have poor emulation capabilities of univariate distributions.

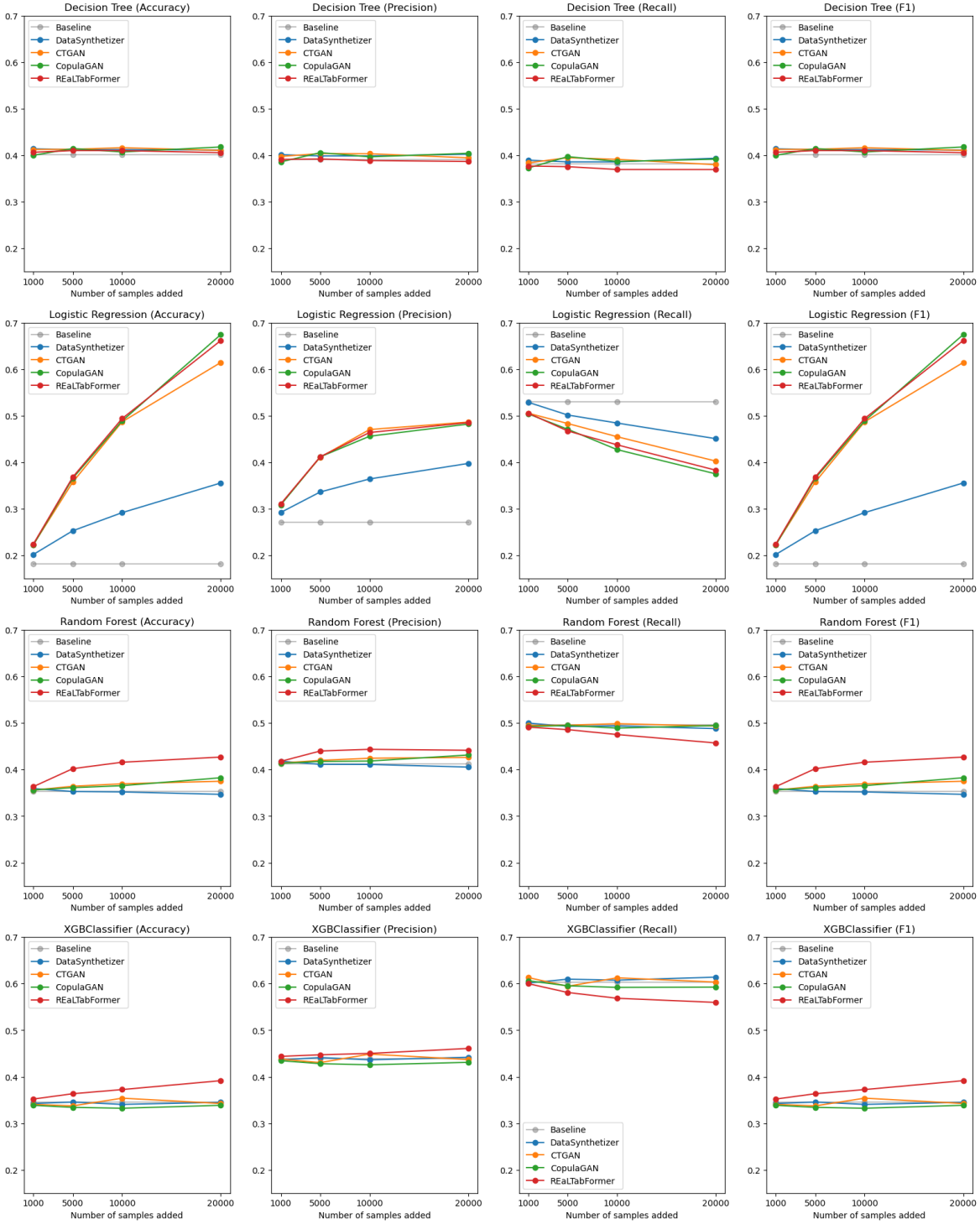

Figure 2. Comparison of the four classification model performance metrics on Class 1 on the test set for each synthesis method used for data augmentation for Adults Income dataset. On the *y*-axis there is the value of each score which can go from 0 to a maximum of 1.

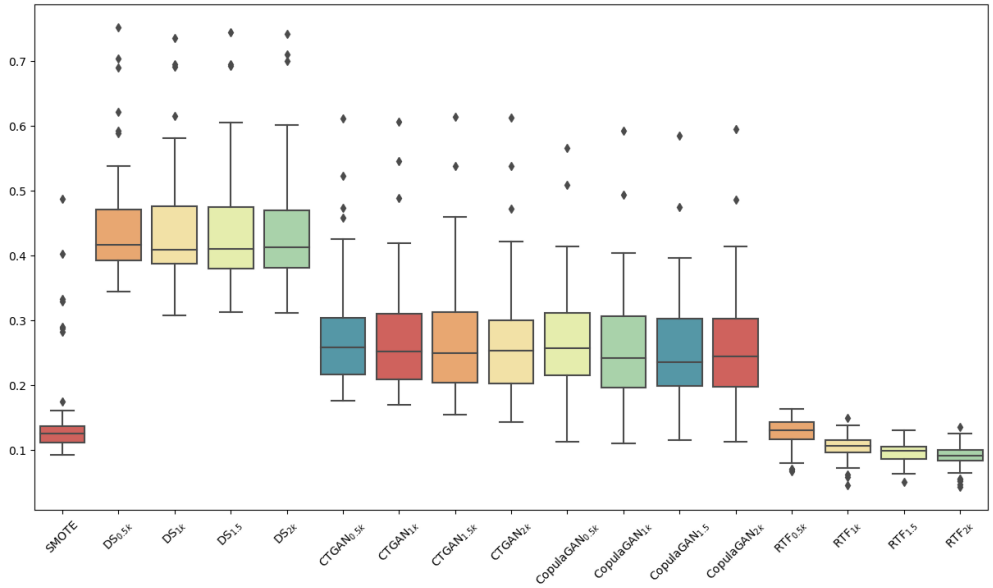

Figure 3. Boxplots of the Jensen–Shannon measures for each synthesis method for the Liar+ dataset. On the y-axis we have the measure computed on each variable of the dataframe.

The PCD follows the same pattern as the Adults Income dataset (Table 4). As the number of samples increases, the relationships among variables are better captured. Borderline SMOTE and REaLTabFormer produce the best-performing synthetic data by demonstrating the ability to capture and then represent relationships component by component. Propensity scores are collected in Table 5. Similar to the Adult Income dataset, the pMSE is highest when the generated sample size is similar to the original sample size (1216), which in this case is 1000. The values exhibit minimal fluctuations based on the generation model used, with the exception of Borderline SMOTE, which outperforms the others. In this scenario, St pMSE values remain consistent within a narrow range regardless of the volume of the synthetic data, unlike the previous case with the Adult Income dataset. This highlights the score's independence from the sample size. Furthermore, since Borderline SMOTE generated 2399 samples, we compare its score with that of the other datasets generated with 2000 samples. It emerges as the model that yields superior values. In terms of classification models, we focused on analyzing the classification models' ability to identify Class 1 data, which is more of our interest as it is the minority class. The F1 scores for Class 1 are presented in Table 6.

Table 4. Pairwise correlation difference in the Liar+ dataset for each synthesis method used.

	PCD		***PCD***
SMOTE	1.6030		
$DS_{0.5k}$	4.3507	$CopulaGAN_{0.5k}$	5.4978
DS_{1k}	4.0252	$CopulaGAN_{1k}$	5.3046
$DS_{1.5k}$	3.8881	$CopulaGAN_{1.5k}$	5.0944
DS_{2k}	3.8144	$CopulaGAN_{2k}$	5.0632
$CTGAN_{0.5k}$	5.5091	$RTF_{0.5k}$	2.9800
$CTGAN_{1k}$	5.1432	RTF_{1k}	2.6132
$CTGAN_{1.5k}$	5.0166	$RTF_{1.5k}$	2.3818
$CTGAN_{2k}$	5.0557	RTF_{2k}	2.3149

Table 5. Propensity score and standardized propensity score for the synthetic data generated from the Liar+ dataset.

	pMSE	***St pMSE***		***pMSE***	***St pMSE***
SMOTE	0.1654	3.2076			
$DS_{0.5k}$	0.1992	3.5757	$CopulaGAN_{0.5k}$	0.2041	3.6839
DS_{1k}	0.2404	3.5669	$CopulaGAN_{1k}$	0.2458	3.6514
$DS_{1.5k}$	0.2391	3.5069	$CopulaGAN_{1.5k}$	0.2462	3.6545
DS_{2k}	0.2294	3.6406	$CopulaGAN_{2k}$	0.2346	3.7681
$CTGAN_{0.5k}$	0.2034	3.6841	$RTF_{0.5k}$	0.2041	3.7280
$CTGAN_{1k}$	0.2434	3.6127	RTF_{1k}	0.2456	3.6599
$CTGAN_{1.5k}$	0.2439	3.6097	$RTF_{1.5k}$	0.2452	3.6772
$CTGAN_{2k}$	0.2315	3.7486	RTF_{2k}	0.2348	3.8048

Table 6. F1 scores in percentage of the classification on the Liar+ dataset computed on Class 1 of the test set for every classification model trained on Class 1 augmented with 2000 synthetic samples.

	Decision Tree	**Logistic Regression**	**Random Forest**	**XGBoost**
Baseline	54.28	25.32	49.23	53.94
Borderline SMOTE	52.85	**72.04**	**62.94**	**61.51**
DataSyntesizer	51.31	40.46	51.42	55.92
CTGAN	53.17	31.58	52.74	56.91
CopulaGAN	53.84	44.41	53.51	53.94
REaLTabFormer	**55.04**	67.11	50.55	54.61

The decision tree (Figure 4, row 1), random forest, and XGBoost exhibit superior performance only when the augmented data are derived from Borderline SMOTE. Conversely, when using data generated by other generative models, these three classifiers do not display any improvement, as we can see in the first, third, and fourth rows of Figure 4, which are the scores corresponding to the decision trees, random forest, and XGBoost models. These rows present the accuracy, precision, recall, and F1 scores from left to right, respectively. Notably, these scores remain unchanged from the baseline (gray line) and show no variation with the number of samples added. This phenomenon can be attributed to the Liar+ dataset's high dimensionality (50 features), making it more challenging to classify accurately. However, logistic regression stands out as the only model consistently showing improvements across all performance scores when class augmentation is applied, as evident from the second line in the figure. Due to its simplicity and heightened sensitivity to class imbalance, logistic regression displays a robust reliance on the number of added samples, leading to enhanced performance across all generative models.

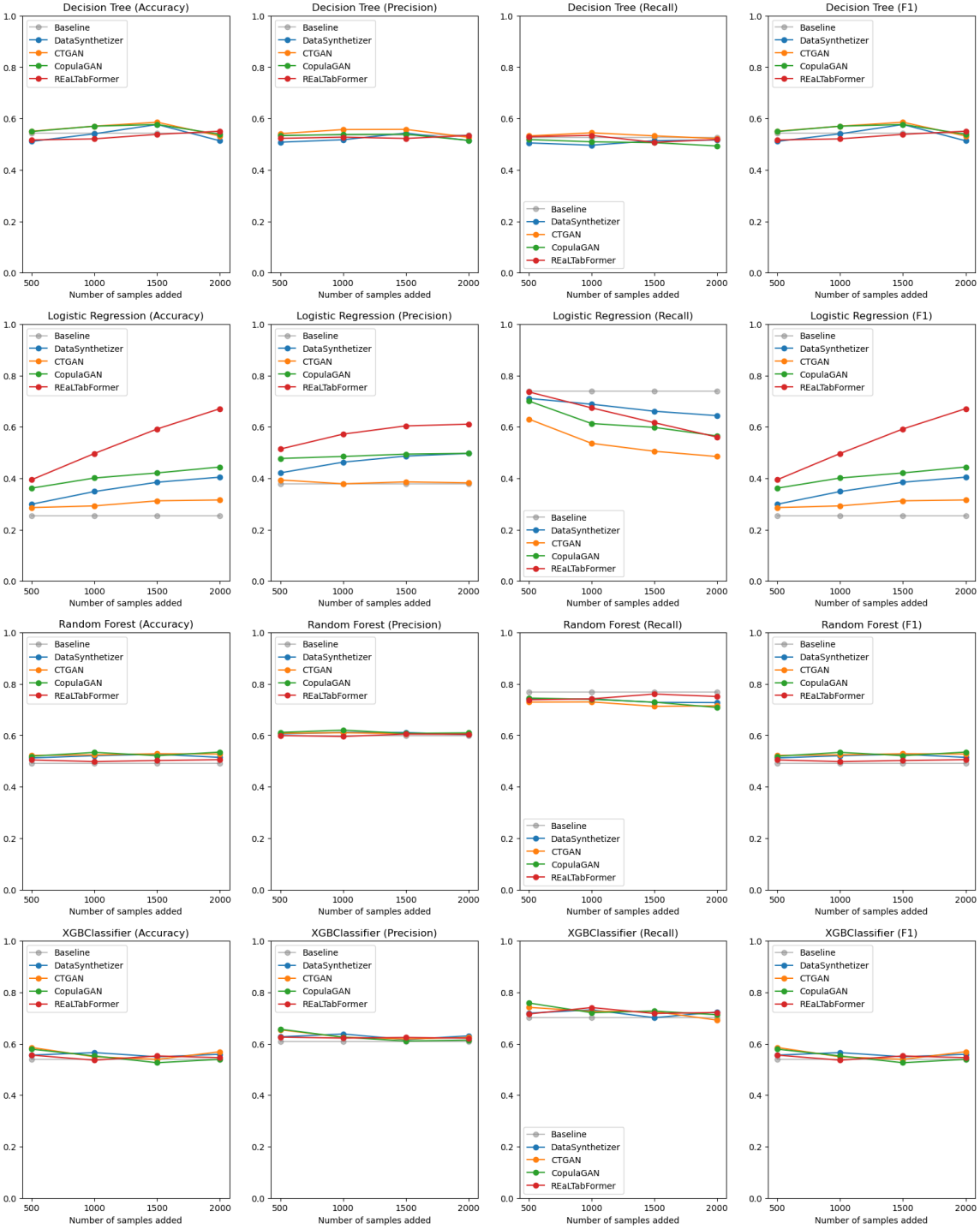

Figure 4. Performance scores on Class 1 of the classifier models on the test set of the Liar+ dataset with varying numbers of synthetic samples from different generation models. On the *y*-axis, there is the value of each score which can go from 0 to a maximum of 1.

In summary, for both datasets, Borderline SMOTE and REaLTabFormer stand out as generative models that exhibit superior performance across various utility measures. Notably, Borderline SMOTE consistently outperforms other models across all evaluated measures, in particular, in terms of PCD (Tables 1 and 4) and pMSE (Tables 2 and 5). The superiority of these models is also evident in the classification results, as all classification models perform better when trained on data generated by Borderline SMOTE and REaLTabFormer (Table 6). Based on the obtained results, it can be concluded that the utility

measures demonstrate a general consistency among themselves. If a generative model excels in one measure, it is likely to excel in other measures as well. Analyzing the data generated by DataSynthesizer, we find that the Jensen–Shannon measure (Figures 1 and 3) indicates a poor univariate match between the synthetic data and the real data as the scores are very high compared to other models. However, the pairwise correlation difference (Tables 1 and 4) and pMSE (Tables 2 and 5) suggest that the bivariate and multivariate distributions are as good as the ones generated by the other generative models. Notably, when it comes to classification tasks, augmenting the data with DataSynthesizer-generated samples leads to relatively less pronounced performance improvement compared to other augmentation techniques, as evident from Figures 2 and 4. The blue line representing DataSynthesizer is consistently positioned below all the other lines in these figures. This indicates that if the univariate distribution of synthetic data diverges significantly from the real data, the other measures related to bivariate and multivariate distributions may have limited relevance in evaluating the quality of the synthetic data. CTGAN and CopulaGAN, with the latter being an improvement over the former, demonstrate comparable performance across most measures, except for a slight advantage of the latter in the classification task. In Figures 2 and 4, the two models correspond to the yellow and green lines, respectively. It can be seen that the green line stands above the yellow line for accuracy, precision, and F1, while below it for recall, which in this case means less bias in the model. REaLTabFormer and Borderline SMOTE emerge as the top-performing generative models across all evaluation criteria, notably surpassing other models in terms of pairwise correlation difference. Overall, the high quality of the generated data translates into superior classification performance.

7. Conclusions

In this study, we have examined various utility measures for synthetic tabular data, exploring their properties and the aspects they can evaluate. Furthermore, we assessed their applicability in the context of classification tasks with imbalanced classes.

To investigate these measures, we employed five generative algorithms: Borderline SMOTE, DataSynthesizer, CTGAN, CopulaGAN, and ReaLTabFormer. These algorithms were evaluated on two distinct datasets: Adults Income and Liar+. Our study reveals that Borderline SMOTE and REaLTabFormer exhibit superior performance among the generative models investigated. Borderline SMOTE adopts a modified approach to SMOTE, generating samples by interpolating borderline data points from the opposing class, while REaLTabFormer utilizes the GPT-2 generative model. The utility measures demonstrate consistency as the dimensionality of the compared distributions increases. Indices that perform well for univariate measures also tend to excel in bivariate and multivariate measures. However, it should be noted that good performance in bivariate and multivariate cases does not necessarily guarantee the same for univariate distributions. Importantly, these measures align with the classification results obtained after augmenting the minority class samples.

In conclusion, the utility measures analyzed in this study provide valuable insights into the quality of generated synthetic data and can serve as informative tools for data analysis. Future research directions may involve extending our findings by incorporating these measures, for example, in the training process of generative models such as CTGAN, with the aim of enhancing their performance.

Author Contributions: Conceptualization, A.F.; Software, E.E.; Validation, A.F.; Investigation, E.E.; Resources, A.F.; Writing—original draft, E.E.; Writing—review & editing, A.F.; Visualization, E.E.; Supervision, A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kaur, H.; Pannu, H.S.; Malhi, A.K. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput. Surv.* **2019**, *52*, 1–36. [CrossRef]
2. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. Artif. Intell.* **2016**, *5*, 221–232. [CrossRef]
3. Weng, W.H.; Deaton, J.; Natarajan, V.; Elsayed, G.F.; Liu, Y. Addressing the real-world class imbalance problem in dermatology. In Proceedings of the Machine Learning for Health, PMLR, Durham, NC, USA, 7–8 August 2020 ; pp. 415–429.
4. Zheng, W.; Xun, Y.; Wu, X.; Deng, Z.; Chen, X.; Sui, Y. A comparative study of class rebalancing methods for security bug report classification. *IEEE Trans. Reliab.* **2021**, *70*, 1658–1670. [CrossRef]
5. Rivera, G.; Florencia, R.; García, V.; Ruiz, A.; Sánchez-Solís, J.P. News classification for identifying traffic incident points in a Spanish-speaking country: A real-world case study of class imbalance learning. *Appl. Sci.* **2020**, *10*, 6253. [CrossRef]
6. Isangediok, M.; Gajamannage, K. Fraud Detection Using Optimized Machine Learning Tools Under Imbalance Classes. *arXiv* **2022**, arXiv:2209.01642.
7. Varmedja, D.; Karanovic, M.; Sladojevic, S.; Arsenovic, M.; Anderla, A. Credit card fraud detection-machine learning methods. In Proceedings of the 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 20–22 March 2019; pp. 1–5.
8. Salah, I.; Jouini, K.; Korbaa, O. On the use of text augmentation for stance and fake news detection. *J. Inf. Telecommun.* **2023**, 1–17. [CrossRef]
9. Vaz, B.; Bernardes, V.; Figueira, Á. On Creation of Synthetic Samples from GANs for Fake News Identification Algorithms. In *Information Systems and Technologies: WorldCIST 2022*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 3, pp. 316–326.
10. Frid-Adar, M.; Klang, E.; Amitai, M.; Goldberger, J.; Greenspan, H. Synthetic data augmentation using GAN for improved liver lesion classification. In Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; pp. 289–293.
11. Jain, S.; Seth, G.; Paruthi, A.; Soni, U.; Kumar, G. Synthetic data augmentation for surface defect detection and classification using deep learning. *J. Intell. Manuf.* **2022**, *33*, 1007–1020. [CrossRef]
12. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.A. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv* **2018**, arXiv:1808.02455.
13. Hernandez, M.; Epelde, G.; Alberdi, A.; Cilla, R.; Rankin, D. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing* **2022**, 493, 28–45. [CrossRef]
14. Assefa, S.A.; Dervovic, D.; Mahfouz, M.; Tillman, R.E.; Reddy, P.; Veloso, M. Generating synthetic data in finance: Opportunities, challenges and pitfalls. In Proceedings of the First ACM International Conference on AI in Finance, New York, NY, USA, 15–16 October 2020; pp. 1–8.
15. Shafique, R.; Rustam, F.; Choi, G.S.; Díez, I.d.l.T.; Mahmood, A.; Lipari, V.; Velasco, C.L.R.; Ashraf, I. Breast cancer prediction using fine needle aspiration features and upsampling with supervised machine learning. *Cancers* **2023**, *15*, 681. [CrossRef]
16. Abd Al Rahman, M.; Danishvar, S.; Mousavi, A. An improved capsule network (WaferCaps) for wafer bin map classification based on DCGAN data upsampling. *IEEE Trans. Semicond. Manuf.* **2021**, *35*, 50–59.
17. Strelcenia, E.; Prakoonwit, S. Improving Classification Performance in Credit Card Fraud Detection by Using New Data Augmentation. *AI* **2023**, *4*, 172–198. [CrossRef]
18. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
19. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223.
20. Doersch, C. Tutorial on variational autoencoders. *arXiv* **2016**, arXiv:1606.05908.
21. Pardo, L. *Statistical Inference Based on Divergence Measures*; Chapman & Hall/CRC Press: Boca Raton, FL, USA, 2005.
22. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
23. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **1991**, *37*, 145–151. [CrossRef]
24. Goncalves, A.; Ray, P.; Soper, B.; Stevens, J.; Coyle, L.; Sales, A.P. Generation and evaluation of synthetic patient data. *BMC Med. Res. Methodol.* **2020**, *20*, 108. [CrossRef]
25. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; Johns Hopkins University Press: Baltimore, MD, USA, 2013.
26. Fasano, G.; Franceschini, A. A multidimensional version of the Kolmogorov–Smirnov test. *Mon. Not. R. Astron. Soc.* **1987**, *225*, 155–170. [CrossRef]
27. Snoke, J.; Raab, G.M.; Nowok, B.; Dibben, C.; Slavkovic, A. General and specific utility measures for synthetic data. *J. R. Stat. Soc. Ser. A* **2018**, *181*, 663–688. [CrossRef]
28. Becker, B.; Kohavi, R. Adult. In *UCI Machine Learning Repository*; Department of Information and Computer Science, University of California: Irvine, CA, USA, 1996. [CrossRef]
29. Wang, W.Y. "Liar, Liar pants on fire": A new benchmark dataset for fake news detection. *arXiv* **2017**, arXiv:1705.00648.

30. Agrawal, R.; Srikant, R.; Thomas, D. Privacy preserving OLAP. In Proceedings of the Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, Baltimore, MD, USA, 14–16 June 2005; pp. 251–262.
31. Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; Dwork, C. Learning fair representations. In Proceedings of the International Conference on Machine Learning, PMLR, Atlanta, GA, USA, 17–19 June 2013; pp. 325–333.
32. Ding, F.; Hardt, M.; Miller, J.; Schmidt, L. Retiring adult: New datasets for fair machine learning. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 6478–6490.
33. Zhang, X.; Ghorbani, A.A. An overview of online fake news: Characterization, detection, and discussion. *Inf. Process. Manag.* **2020**, *57*, 102025. [CrossRef]
34. Kaliyar, R.K.; Goswami, A.; Narang, P. FakeBERT: Fake news detection in social media with a BERT-based deep learning approach. *Multimed. Tools Appl.* **2021**, *80*, 11765–11788. [CrossRef] [PubMed]
35. Nasir, J.A.; Khan, O.S.; Varlamis, I. Fake news detection: A hybrid CNN-RNN based deep learning approach. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100007. [CrossRef]
36. Vaz, B.G. Using GANs to Create Synthetic Datasets for Fake News Detection Models. Master's Thesis, Universidade do Porto, Porto, Portugal, 2022.
37. Han, H.; Wang, W.Y.; Mao, B.H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In Proceedings of the Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, 23–26 August 2005; pp. 878–887.
38. Ping, H.; Stoyanovich, J.; Howe, B. Datasynthesizer: Privacy-preserving synthetic datasets. In Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, 27–29 June 2017; pp. 1–5.
39. Xu, L.; Skoularidou, M.; Cuesta-Infante, A.; Veeramachaneni, K. Modeling tabular data using conditional gan. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 7335–7345.
40. Copula GAN Synthesizer. Available online: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/copulagansynthesizer (accessed on 17 March 2023).
41. Solatorio, A.V.; Dupriez, O. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers. *arXiv* **2023**, arXiv:2302.02041.