CREATING SYNTHTIC DATA TO HANDLE CLASS IMBALANCE

Avi Gupta
School of Computing
Newcastle University, UK
a.gupta7@newcastle.ac.uk

Professor Paolo Missier
School of Computing
Newcastle University, UK
paolo.missier@newcastle.ac.uk

ABSTRACT

Synthetic data is the beginning of a new era in the world of data science where it can reduce our data requirement and can handle bias in a dataset. But the generation such data which can be created on demand is a difficult task. In this study, we will try to generate synthetic versions of 2 categories of dataset. The first group of sets will follow the single tabular format in which four publicly available datasets will be used whereas the second band will adhere to the relational setup. The study will be conducted in two phases. The first phase will be the generation of synthetic set in which four pre-developed synthetisers will be used that are available as libraries of the synthetic data vault package whereas in the second phase, we are going to evaluate the data quality produced from these generators. The evaluation will be done on two metrics which are accuracy-based metric & statistical metric.

Keywords:- Copula, GAN, Propensity Score, Chi-Squared Test, Kolmogorov Smirnov Test, Logistic Regression

1 Introduction

We live in a world where data is said to be the new currency in our daily life. With the evolution and increase of demand in the technology, our dependency of data has also increased. [2]. The field where data act as an essential fuel is artificial Intelligence and machine Learning. According to a research it was found that AI & ML is going to be the new powerhouse for the business and a tool of improvement in our daily life. [4]. But for developing an AI or a ML model we need high quality of data in huge amounts. [3] and the collection of such data is another task. Even if we are able to gather the required data, the next problem that will act as a hurdle will be the imbalanced class within that dataset. Imbalance class is a problem in which the distribution of the categories in a dataset are skewed. [17]. This is where synthetic data steps in. The most exciting thing about synthetic data is that it not only solves the issue of class imbalance but can also be used as tool in the matter of data privacy. [1]. It is not audacious to say that a synthetic version can also be referred to as an alternate version of the original dataset. The synthetic data can also be used for finding the biases in those data that is being used as a basis for developing some algorithms for machine learning models. [5]. Using synthetic data is not only going to help in securing user's data but it can also solve the important issue of biases in a dataset. Before going further, it is important to know the meaning of "bias in a dataset". The term bias in the field of data is considered to be a type of error in which particular datapoints in a dataset are given more priority as compared to rest of the elements. In other words, these datapoints are presented more in the dataset. [3]. It is said that a biased dataset is terrible for a machine learning models as using these kind of datasets in any machine model can result in that model's low accuracy levels. Apart from these models, biased dataset can also be proved as a reason for error in analytic result and predictive outcomes. [6]. The main question that arises is what is synthetic data? How is it generated? and how it can resolve all of the mentioned issues. To answer that first we need to focus on the working of a predictive algorithm. In summary a predictive algorithm tries to figure out the statistical feature within the dataset and based on the discovered statistical feature it provides a value. For example, a classification model's algorithm such as Logistic regression finds the probability of a data point being under a particular category. [7].

Now, focusing on the definition of Synthetic data, Synthetic data is nothing but the creation of fake data which is manufactured mathematically by using a statistical model. Considering machine learning models, the synthetic data will preserve the same statistical feature as the original dataset. In the case of data privacy, the generated data will be separating the sensitive information from its original source. Another interesting aspect of the synthetic data is that it can also enlarge a dataset by recognizing the statistical trends from the entire dataset and then generating the synthetic data points using the identified statistical feature. This technique of expanding the dataset can also improve the precision of the analytics result from that dataset. [8].

This study will focus on using the gaussian copula technique which belongs to the multivariate copula family. [13]. Apart from copula methods, another generator that will be used for the experiment in this paper will follow the GAN based approach. [23]. The next synthesizer in this study will be a combination of copula & GAN which will follow the footsteps of cumulative distribution function-based transformations. [27].

Main aim of this study will concentrate on using the above-mentioned synthesizers to generate synthetic instances and evaluating their quality on the basis of classification accuracy & statistical metrics. Apart from these metrics, one synthetic dataset will be evaluated on analytical similarities with the original set. The paper will be divided into 4 sections. The first section will highlight the related literature that has been researched. Second section will discuss about the experiment that has been conducted to fulfil the aim of this study. Third section is about the results that is obtained from the experiment whereas fourth section will talk about the limitations that has been discovered in the methodology throughout the experiment and the work that can be done in future.

2 Related Work:

The purpose of this segment will concentrate on the literature which displays the work that has been done till now regarding the generation of the synthetic dataset. The research was conducted by considering three aspects. The first part will display some of the techniques and generators that are able to produce synthetic data. In the second aspect, quality of the generated data will be discussed. In the third we are going to discuss about an experiment that resulted in the generation of synthetic dataset.

2.1 Synthetic Data Generators

The first method for generation is known as data synthesizers and was created in 2017 by using the programming language python. This generator extracts the correlation in a dataset by using *Bayesian network* technique. [15]. Bayesian network is a graphical model that captures the joint distribution for a set of variables. [18]. This approach can be applied on those datasets that are in the numerical, datetime, categorical, and non-categorical string format. Data synthesizer is also equipped with the attribute of converting the missing values in the dataset by taking the rate at which the values are lacking. [15, 17].

Next synthesizer in the line was developed under the name of Synthetic Data Vault in 2016. This synthesizer also captures the joint distribution in the dataset by using *gaussian copula* model. This model is utilized for highlighting the marginal and cumulative distribution in the dataset. [14]. The SDV works on the assumption that the provided dataset is in the numerical form. When this assumption is not fulfilled the algorithm of the model applies some basic data pre-processing which converts the categorical data points into numerical which stays in the range of 0 & 1. [11]. The joint distribution is referred as the distribution of a dataset against the distribution of another set of data points whereas the marginal distribution can be considered as the summation of the row or the column in a dataset. [9]. As for the cumulative distance, it is the total of the occurrence of the points in a dataset. [10].

Third and fourth generators in the pipeline is known as Synthpop which was created using R programming language in 2016. [16]. This synthesizer captures the conditional distribution from the dataset and then generates synthetic dataset sequentially. [11]. The synthpop or SP uses two techniques to capture the conditional distribution. The first approach is known as SP nonparametric or *SP-np* in which the classification and regression tree algorithm is used whereas the second procedure works on the logistic & linear regression algorithm. This is described as the parametric approach. [35].

The fifth generator is based on deep neural network. The main focus of this synthesizer is targeted on using a GAN based generator. GAN or Generative Adversarial Network is able to provide a way that can help in the learning process of deep representation without bothering about the annotation condition of the training data. To achieve such complexity, GAN conducts the process of backpropagation through the competitive process that includes a pair of deep neural network. [12]. This fifth generator is known as *CT GAN*. It contains the ability to produce synthetic data points based on the distribution of a tabular dataset. This model follows the *mode-specific normalization* that can overpower those statistical distribution that are not able to recognize by Bayesian & copula-based generators. [23].

2.2 Utility Measures

After obtaining synthetic data generators, next step in the research journey is to find the means of evaluating the data quality that these model produce. The first measure in the line-up is referred to as the *propensity score.* [19, 20]. This scoring system is the representation of the probabilities of the data points that appears in the original and synthetic datasets. To obtain this score the first task is the pre-processing of the datasets where the synthetic edition is merged with the original set along with adding a variable such as *T* where the initialized variable will be equal to one for those points that are from the synthetic version and zero will be given to those data that belongs to the original set. [22]. After initializing the values, the second job will entail the use of a binary classification model to distinguish between the synthetic and original instances. The results that will be obtained from this model will be used to calculate the propensity score with the help of this formula:

$$pMSE = {}^{1}/_{N}\, \Sigma(\hat{\rho}i - 0.5)^2$$

Here, $\hat{\rho}$ is the propensity score, i is the record in a dataset, and N is the size of the dataset,

The value that will obtained from the above-mentioned formula will range between 0 and 0.25, where 0 will indicate that the synthetic point is indistinguishable from the original data. The case where the score value is 0 will only occur when the generator is in the scenario of overfitting.

The next evaluation metric in the pipeline is not a metric but a technique that has been proposed by Dalianis H. This technique is known as Train on synthetic & test on real or vice versa. The first step in this metric includes the training & testing of any machine learning classification model on a real dataset with which a benchmark score can be established. Second step involves the generation of synthetic form. After this stage we can either train the machine learning model that was used to get the benchmark score on synthetic data and then test it on real set or we can use the already trained model and test it on the synthetic instances. This technique can help in the process of evaluating synthetic set that involves a classification task. [24].

Apart from the above mentioned, the synthetic data vault library also comes with their own evaluation metric library which is known as *evaluate* in which some statistical test such as *chi-squared test, two sample Kolmogorov test, gaussian mixture log likelihood* , and many more can be calculated. After the scores are obtained the function takes an average of the calculated values and as a result the function returns a range of numbers between 0 – 1. The indication of the quality in this metric depends on the obtained value proximity to 1 where 0 is designated as the worst quality and 1 as the best. [25].

Following the *evaluate* function, another form of metric that was found is known as the hypothesis test. Hypothesis testing is recognized as a statistical method in which a set of claim is taken into consideration by the researchers. These claims in technical term are known as *null hypothesis* and *alternate hypothesis*. *Null Hypothesis* (H_0) suggest that there is no significant statistical difference between two population whereas the *alternate hypothesis* (H_a) recommends that there is a considerable statistical disparity between the samples. These statistical characteristics are defined according to the type of statistical test that is conducted in order to calculate the *p-value. P-value* facilitates the researchers in choosing between the claims. [26]. The threshold for the *p-value* is 0.05. If the resulting *p-value* is greater than 0.05, in that case we can accept the null hypothesis and vice-versa.

From numerous statistical test, there are two test which comes under the umbrella of non-parametric statistical analysis and are suitable for synthetic dataset evaluation. The first test is known as the *chi-squared test.* This test is conducted to verify if there are noteworthy differences between the normal and observed frequencies in two or more than two data samples . [33]. In terms of evaluating synthetic data, *chi-squared test* can be used to identify the distributions from the original & synthetic dataset and by using hypothesis testing where the *p-value* will be calculated to determine the distribution similarity between both sets as part of the evaluation. [25]. The calculation of *chi-squared value* can be accomplished with the help of this formula:

$$X^2 = \Sigma\, \frac{(Oi - Ei)^2}{Ei}$$

Here, X^2 is the chi-squared test value, Oi are the observed frequencies, and Ei are the expected frequencies. In this pipeline, the second test is referred to as *two-sample Kolmogorov Smirnov test.* In layman terms this test is conducted to see whether two samples follow the same empirical distribution or not. This test is also conducted as a part of hypothesis testing where *null hypothesis* (H_0) states that there is no notable difference whereas the *alternate hypothesis* (H_a) contradicts the claim. The formula for calculation can be found in [34].

2.3 Experiment & Result

Till now we have discovered various pre-developed models that possess the ability to extract significant statistical distribution from the original dataset and based on the obtained distributions generating a fresh set of data points. Along with the models we have also highlighted some of the evaluation metrics that are able to tell us the quality of the generated data. Now, in this section we are going to see an experiment that was conducted on a number of datasets and the obtained result that showcased the quality of the generated synthetic data.

This experiment was conducted by Dankar and Ibrahim in 2021 [35]. Here, 15 datasets were considered. Description of each dataset can be seen in ***Appendix 1.*** This experiment was conducted in a three-step procedure. The first step included the random splitting of the original dataset into 70:30 where 70% belonged to the training set and 30% to the test set. After data separation, four synthetic data generators that are **data synthesizer**, **synthetic data vault**, **synthpop parametric**, & **synthpop non-parametric** were applied. These generators are used repeatedly in each step for all the datasets. The second step included the calculation of the propensity score where the score was computed for each synthetic dataset. In the last step, prediction accuracy was also computed as one of the evaluation metric where the technique of *train on synthetic & test on real* was used. The detailed version for each step can be located in ***Appendix 2***. The result of this experiment concluded that the synthetic data should be generated on a raw dataset ***Appendix 3***. The next reflection of the experiment states that sharing the same parameter tuning of real set with the synthetic version can improve the prediction accuracy score. ***Appendix 4.*** Another observation made by this experiment suggest that in terms of using the prediction accuracy as one the evaluation metric, that changing the parameter settings on machine learning model will not create a significant change in the accuracy score ***Appendix 5***. Final suggestion of this study is regarding the use of propensity score and prediction accuracy score. According to the experiment result it is believed that propensity score is only good when **data synthesizer,** , **synthpop parametric**, & **synthpop non-parametric** are used. In the case of **synthetic data vault** prediction accuracy score is proved to be a better evaluation metric. ***Appendix 6***.

3 Methodology

The next stage of this paper will concentrate on creating an experiment framework where synthetic data generation will be done with the help of pre-developed generators. The experiment will work on two folds:

- Providing a proof of concept that using a pre-developed generator we can produce synthetic data that will resemble the same statistical features as the original dataset.
- The second fold will focus on the evaluation of the generated points. The goal here will be to use some evaluation metrics or techniques with which the quality of the generated data can be tested and can inform whether merging the synthetic instances will have the positive or a negative influence on the efficiency of the original dataset.

This segment will be further divided into five components. Section 3.1 will contain a detailed information about the datasets that has been used in this experiment. Section 3.2 will discuss about the pre-developed data generator models that has been used in this study for the generation of synthetic data. Section 3.3 is about the evaluation metrics that has been used in this study. Section 3.4 will display the information about the data pre-processing step that was imposed on the selected dataset & the experimental setup to generate synthetic points.

3.1 Data:

For the purposes of this experiment, five datasets have been taken into consideration. These five datasets are divided into two categories of tabular datasets, **Single tabular dataset** & **relational dataset**. Under the umbrella of single tabular category, 4 datasets have been used. These datasets are classified as follows:

- **38_Sick:**
  This dataset contains a binary classification task in which two categories have been identified. The identified category is named as **negative** & **sick**. The dataset contains 3018 rows and 30 columns. These 30 columns contain the information from patient's age to their pregnancy status. There are two columns with integer data type along with five columns that contains the value of real number datatype whereas the rest of the column is of categorical data type. Apart from these details regarding the dataset structure, it also contains some empty values or in technical terms NULL values which was removed during the pre-processing phase. This dataset contains the information of certain test with which the classification of a patient being "sick" or "not sick" has been made. The dataset is publicly available with detailed information at [28].

- **185_Baseball:**
  This is multi class problem in which categories are divided into three numbers that are **0, 1, 2**. This dataset comprises of 1073 rows and 18 columns. In these 18 columns three columns contains data points of categorical data type whereas only four columns contain real numbers as data and the rest of the columns are of integer data type. Like the previous dataset this too contains some empty rows. Here, dataset holds the information of a player's statistics that are contributing of them being in the hall of fame. Information of these statistics can be found at [29].

- **LI0_1487_Ozone_Level_8hr:**
  The next dataset in the pipeline is again contains the binary classification-based task. In this dataset the classes are labelled

as **1** & **2**. It consists of 2028 rows and 73 columns in which only one column is of integer type and one column which contains the classification information of the dataset is of categorical type whereas the rest of the columns is of integer data type. Unlike the previous two datasets, this set does not contain any NULL values. This set contains the information about the levels of ozone layer that has been collected in time frame of 8 hours. Detailed report about the feature variables can be found at [30].

- **UKDA – 8722:**
  This single tabular dataset is available in SPSS format which stands for **S**tatistical **P**ackage for **S**ocial **S**ciences which contains certain packages that are useful for exploratory data analysis when a researcher is dealing with scientific data that is associated to social sciences. [31]. This dataset does not contain any regression or classification. It consists of 3224 rows and 572 columns in which some of the columns contains NULL values. This dataset is the representation of the survey answers that highlights the lifestyle of the population and was conducted in 2019. This dataset contains NULL values in certain columns.

The second category in this experiment pipeline is relational dataset in which only one set was taken into consideration:

- **32_Wikiqa:**
  It is a WikiQA relational dataset that contains a set of questions and answers [32] in which the classification of all the sets are as follows:

  - **learningData** which consist of 23406 rows and 3 columns. This part of the dataset is used for fulfilling the classification task. Apart from the structural details of the dataset, it is also being used as the parent dataset where column **qIndex** & **sIndex** are the primary key which is behaving as relationship link between rest of the datasets.
  - **questions** consist of 3047 rows & 2 columns in which the primary column is **qIndex**.
  - **Sentences** consist of 26196 rows & 2 columns in which the primary column is **sIndex**. [32].

3.2 Models:

In this study three data generators or in other terms pre-developed models have been taken into consideration. The classification of these models are as follows:

Model 1: Gaussian Copula:

Gaussian copula is part of copula family. Copulas are classified as the statistical models that are able to identify the dependencies structure amongst various statistical distributions. [36]. Copulas were first introduced by Sklar [37]. In this study, a pre-developed gaussian copula model has been used that was imported from the synthetic data vault library. This pre-developed model is created with the help of multi column gaussian distribution which is also known as normal distribution or joint distribution. Gaussian copula models the univariate distributions and generalizes them to multivariate columns by using the technique of probability integral transformation where the function is working on an assumption that if X is a random variable having the cumulative function distribution f_x then another variable Z will be denoted as $Z = f_x(X)$. [38].

Mathematically, Gaussian copula can be expressed as $C_{\Sigma}^{Gauss}(u) = \Phi_{\Sigma}(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_d))$ where Φ denotes the cumulative distribution function of gaussian or normal distribution. [36].

Model 2: Conditional GAN(CT GAN):

Next generator in the pipeline is based on the GAN model which has been discussed in section II A. In a brief, tabular data is a complex structure where the combination of continuous & discrete variables are present which makes it difficult the process of modelling because sometimes discrete columns are imbalanced making it difficult to use a basic statistical or deep learning model. To overcome these flaws the concept of CTGAN was introduced which utilizes the conditional generators to produce synthetic instances.

For this study, the model was imported from the tabular library of the synthetic data vault package in python. The imported model used the technique of normalization which helped in overpowering the non-gaussian based distribution. Apart from the mode specific normalization methods, the model uses the conditional based generators along with sample training which helps in boycotting the issue of imbalanced discrete variables. The model also uses the fully connected layer structure in which it is using the activation function of *leaky ReLu* & *mix activation functions*. For normalization, the model is working on three step method. The first step includes the user of variational gaussian mixture which is estimating the mode number and learning the gaussian mixture distribution. Second step includes the computation of probability for each value in a continuous column coming from each mode. Last step in the process incorporates the sampling of mode according to the intensity of the computed probability and using the sampled mode to conduct the normalization of the value. To summarize, the structure of the model, fully connected layers has been used in both generator & which are deep reinforcement learning technique and are categorized as:

- For generator, ReLu activation function is being used for conducting normalization task along with two dense layers after which row representation of synthetic instances are produced with the help of mix activation function. Scaler values in this technique is generated with the help of tanh whereas mode indicators are generated by using SoftMax function.
- In critic, instead of mix activation function, leaky ReLu is being used for hidden layers along with a dropout layer. [23].

Model 3: Copula GAN:

This model is a combination of gaussian copula & CTGAN model in which cumulative distribution function-based transformers are used along with CTGAN model to highlight the statistical distribution in the dataset. [27]. For this study the model was again imported from the tabular library of the synthetic data vault package in python.

Till now all the models that have been discussed were used on single tabular dataset. For relational dataset another model was discovered which is referred to as,

Model 4 : Hierarchical Modelling Algorithm Class (HMA1 Class):

Hierarchical modelling algorithm helps in modelling the synthetic versions of relational dataset by recursively walking through the metadata of a relational set and applying gaussian copula model on those relationship fields that's help in generating synthetic version that follows the same statistical traits. [43]. In this study the model was imported from the relational library of the synthetic data vault package.

3.3 Evaluation Metrics

To evaluate these models in this study, certain evaluation measures were taken into which are classified into five metrics:

3.3.1 Train on Real Test on Synthetic:

This technique follows the method proposed by Dalianis H but here instead of training the classification model with the synthetic data points, the model was trained by using a chunk of training set obtained from the original dataset and then the model was used to predict the values by using synthetic set. Before testing the model with synthetic instances, a benchmark accuracy score was obtained by using the *k-fold* validation method. The detailed version of the process can be found in section III(D).

3.3.2 Evaluate:

In this measure, a pre-defined function of synthetic data vault was used in which the method conducts 29 statistical test after which an average of all the scores is calculated. The obtained average lines between the range of 0 to 1. The quality of synthetic point is estimated by the proximity of the averaged value to 1. Details of all 29 test can be found at [7].

3.3.3 Chi-Squared Test:

As discussed in section 2.2, this statistical test helps is check the normal distribution similarity between original & synthetic datasets. This similarity check is done by conducting a hypothesis test in which the null hypothesis states that there is no difference between the distribution of the datasets. The test score is in the form of a *p-value* in which if the value is greater than 0.05, then we can accept the null hypothesis.

3.3.4 Kolmogorov Smirnov Test:

As discussed in section 2.2, this test is conducted to check the similarity of empirical distribution between original & synthetic test. Just like chi-squared, this test also works by performing hypothesis testing where the null hypothesis states that there no significant differences.

3.3.5 Analysis Specific Evaluation:

In this measure, a random EDA is conducted on the original set and then on the synthetic dataset. the goal here is to see whether the synthetic version is able to generate same analysis result as the original set or not.

3.4 Data Pre-Processing & Experimental Set-Up

This section will discuss about the pre-processing step that was conducted before applying the above-mentioned models on the datasets.

3.4.1 38_Sick:

As mentioned in the dataset description, this set includes some NULL value which is why the first step was to see the total number of empty values and it was achieved with the use the pre-defined function of python known as *dataset.isnull().sum()* which returned the number of empty values in each columns which are:

- "sex" -> 125,
- "TSH" -> 295,
- "T3" -> 604,
- "TT4" -> 184,
- "T4U" -> 326,
- "FTI" -> 324, and
- "TBG" -> 3018.

According to this result, "TBG" does not contain any value is as the total number rows in this dataset is 3018 due to which the next step includes the elimination of this column before removing rest of the NULL values. Apart from "TBG" one more column was removed as it contained only the indexing of the dataset. That column is named as "d3mIndex" To remove the "TBG" & "d3mIndex" columns, a python function named *drop()* in which these parameters were passed After removing these two columns, the third step includes the removal of all the NULL values which was done with the help of *dropna()* with some changes in the parameter. After removing all the NULL values, the obtained dataset was passed into the pre-developed synthetic data generators. When the generators were trained, the learned weights of the models were saved.

The next few steps of pre-processing were conducted for the evaluation purposes. In these, the first step included the extraction of predictor and response variable by using the function of *iloc* after which the imbalance of the dataset was checked by plotting a count chart which displayed the total number of data points in each class. The plot can be seen in ***Appendix** 7*. As per the result there was some significant difference between the numbers which why the technique of SMOTE (Synthetic Minority Oversampling Technique) was used. This technique is a part oversampling method in which synthetic points for the minority category are inserted alongside the border of nearest neighbor of minority class instance. [35]. This technique was applied by importing a function named *SMOTE()* from the oversampling class of imblearn in python. The result of this technique in the class imbalance can be seen in ***Appendix 8***. The next step was to prepare the extracted predictor & response variable from the original dataset by encoding them. This task was achieved by using *get_dummies()* function for predictor variables and *LabelEncoder()* function for response variable. This step was done before applying the SMOTE function. After data-pre-processing, a machine learning classifier was trained on the real dataset.

3.4.2 185_baseball:

In this dataset only one column "Strikeouts" was filled with 18 NULL values. These values were removed by using the same technique that was conducted in the '38_Sick' dataset. After removing empty rows, the processed dataset was fitted in all the models. The weights were saved after the generators were trained.

3.4.3 LI0_1487_Ozone_Level_8hr:

Unlike the previous datasets, this set does not have any empty rows which is why only the extraction of predictor & response variable were performed along with the use of SMOTE to resample the underpopulated class. The frequency of categories before & after conducting SMOTE can be seen in ***Appendix 9*** & ***Appendix 10***. Here, after pre-processing, the same classification model that was used during the exploration of '38_Sick' dataset was trained with the original data points.

3.4.4 UKDA - 8722:

Unlike all the previous datasets, only gaussian copula generator was used. In this dataset, instead of eliminating the empty rows, empty columns were removed. Here, 407 out of 572 columns contained empty fields. After removing the columns, the processed set was passed as a parameter in the model.

3.4.5 32_Wikiqa:

In this dataset, a table of metadata containing the structural information about the parent dataset constructed. At this stage primary key columns were also initialized which was later used as a link between the parent and child datasets. The next step was the initialization of the child dataset in which the structural information was added in the existing metadata table containing the information of the parent dataset. After creating a relationship link between all the sets, a dataset dictionary was formed in which the new structure containing the definition of the relationship of the entire 32_wikiqa dataset was given as the input. This dictionary was later passed in the HMA1 generator. In this dictionary on of the set was framed for the classification task which is why the same classifier used in '38_Sick' & '185_Baseball' was trained with the set's original data points.

4 Results

In this section we are going to discuss about the results that has been obtained. These results are classified according to the evaluation metrics that has been used.

4.1 Train on Synthetic Test on Real

For this technique logistic classifier was used in which the first task was to get the classifier's accuracy on the original dataset. This accuracy score was used as the benchmark score and was obtained with the help of *K-Fold* validation technique. In this technique a dataset is divided into k folds in which the model uses each fold as the test set. [26]. In this case the number of used for every datasets were 10. After acquiring the benchmark score, the next step in the evaluation pipeline was applying the same pre- processing steps that was conducted on the original dataset to synthetic version and then using the same trained model and testing it on the synthetic points. The calculation for obtaining the accuracy score on synthetic set was done with the help of *accuracy_score*() function. The results are displayed in table 1:

Table 1:

Dataset	Benchmark Score	Synthetic Data Generators			
		Gaussian Copula	CT GAN	Copula GAN	HMA1
38_Sick	**91%**	**74%**	**84%**	**79%**	**N/A**
LI0_1487_Ozone_Level_8hr	**84%**	**75%**	**78%**	**73%**	**N/A**
32_Wikiqa	**95%**	**N/A**	**N/A**	**N/A**	**2%**

According to the results it can be seen that in 'LI0_1487_Ozone_Level_8hr' in which no data pre-processing was conducted, the accuracy score is close to the proximity of their benchmark results whereas in '38_Sick' where NULL values were removed, the scores are not far but pales in comparison with the rest of datasets. In the case of '32_Wikiqa' the score is not even in boundary of the benchmark proximity. To summarize, it can be concluded that these three generators are able to mimic the probability distribution of the dataset apart from HMA1.

4.2 Statistical Metrics:

In the context of statistical measures, chi-squared and Kolmogorov Smirnov Test were conducted. These test were conducted with the help of a pre-defined function of the SDV package which is known as *evaluate*() in which 'CSTest' & 'KSTest' are passed as the parameters. Apart from these test the function is also able to calculate a score for the quality of the generated dataset. The table below contains the *p-value* which is calculated by subtracting one form the obtained scores from the test.

Dataset	Statistical Metrics										
	Gaussian Copula			CTGAN			CopulaGAN			HMA1	
	CS Test	KS Test	Evaluate	CS Test	KS Test	Evaluate	CS Test	KS Test	Evaluate	CS Test	KS Test
38_Sick	0.24	0.20	0.580477	0.09	0.25	0.604864	0.41	0.28	0.448113	N/A	N/A
185_Baseball	0.000029	0.12	0.578635	0.000016	0.22	0.395106	0.000086	0.31	0.355881	N/A	N/A
LI0_1487_Ozone_Level_8hr	0	0.09	0.565483	0	0.21	0.342457	0	0.19	0.357940	N/A	N/A
UKDA – 8722	0.13	0	0.515714	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
32_Wikiqa	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	0.6

According to these results, it can be seen that considering the '38_Sick' dataset, the best performer amongst all three generators us CTGAN. According to the *p-value* it can be concluded that the synthetic data produced from all three models are fulfilling the statement of null hypothesis but comparing the average score of the evaluate function, CT GAN generator has produced some significant results regarding the synthetic data quality in an overall statistical metric whereas in the case of rest of the datasets not all of the synthetic sets are able to mimic both of the distributions. In the case of '185_baseball' only empirical distributions are able to follow which is the same case for 'LI0_1487_Ozone_Level_8hr'. To summarize, these generators are not able to model gaussian and normal distribution properly from the original dataset.

4.3 Analysis Specific Metrics:

In this metric the result will be compared according to the generator's ability to mimic the same analysis result as it was obtained from the original dataset. For this metric only one dataset has been considered because this evaluation technique is an experiment which is conducted to see whether synthetic dataset can be used as a substitute to the original set or not.

38_Sick:

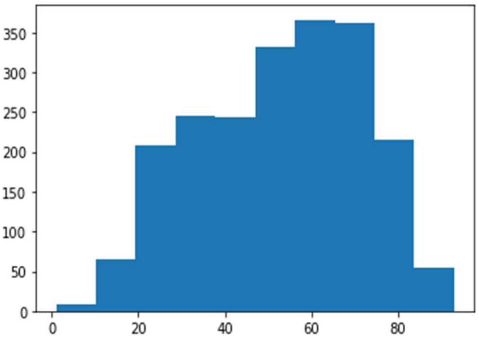


Figure 1: Age range of the patients from the original dataset

According to the figure 1 it can be seen that the patient's ranging from 60 – 75 are high

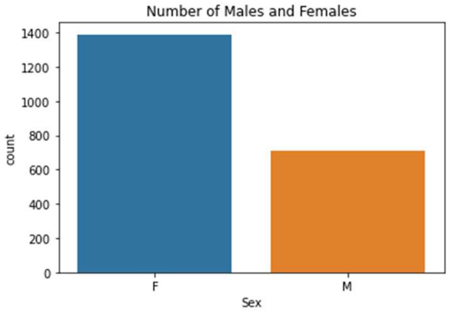


Figure 2: Number of Males & females from the original dataset

Figure 2 proves that female patient shares a large amount of distribution than male.

- **Gaussian Copula:**

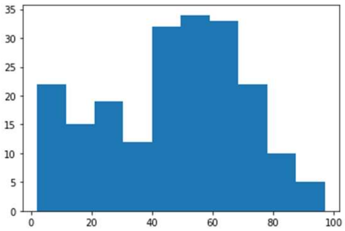


Figure 3: Age Range Distribution Generated from Gaussian Copula

As per the distribution in figure 3 it can be demonstrated that the highest age range is between 40 – 70 which is not far from the proximity of the analysis generated from the original dataset whereas the rest of the age range distribution seems to be very variated.

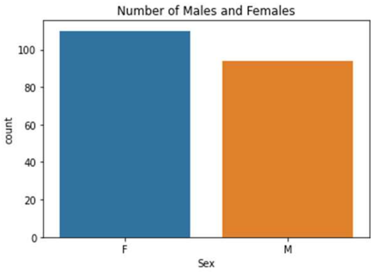


Figure 4: Male & Female Frequencies Generated from Gaussian Copula

In the case of the male and female ratio distribution, the difference of frequencies between the original & synthetic dataset is also significant although according to figure 4 the population of male is less than the female which was similar in the case of original dataset.

In conclusion, the replication of the analysis is not successfully achieved by this generator.

- **CT GAN:**

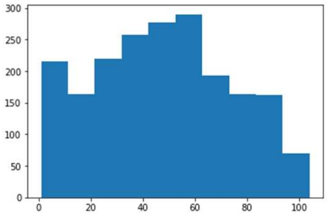


Figure 5: Age Range Distribution Generated from CT GAN

According to figure 5, the age range distributions acquired from the CT GAN generated data contains a considerable variation as compared to the original dataset. Here the highest age range is between 35 – 60.

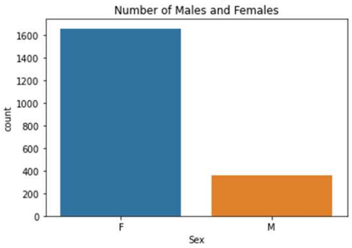


Figure 6: Male & Female Frequencies Generated from CT GAN

As for the male and female population frequencies which is displayed in figure 6, there is a huge difference between the population ratio which is not the case for original dataset.

In summary, the analysis-based generation has not been successful for CT GAN data synthesizer.

- **Copula GAN:**

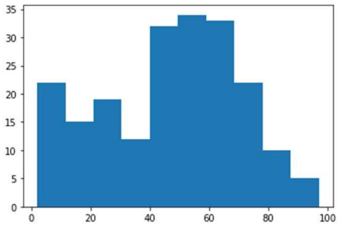


Figure 7: Age Range Distribution Generated from Copula GAN

According to the age range generated from the copula GAN data, the maximum age lies between 40 – 75 which again is not that distant from the original set.

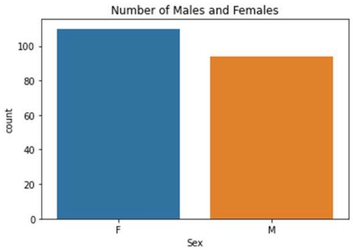


Figure 8: Male & Female Frequencies Generated from Copula GAN

As for the gender frequency, the variation is significant which can also be seen in figure 8.

In conclusion, the result of copula GAN is similar to gaussian copula but overall, this generator also fails in the process of analysis replication.

5. Conclusion & Future Work

Throughout this project we have seen the importance of synthetic data in the real world and how they can be created. We have also discussed about some the pre-developed synthesizers that are capable of producing synthetic versions in which the statistical element of the original dataset can be seen. Now the main question of this section is to see how successful the experiment conducted in section 3 has been in fulfilling the ultimate goal of this study which is to see whether a perfect synthetic version can be generated or not. To summarize this study, The experiment work has been done with four synthesizers which were evaluated on three measures, Machine Learning Classification Accuracy Metric, Statistical Metrics, & Analysis Specific Metrics. According to the experimental results, the data synthesizers were able to mimic the probabilistic distribution of the dataset which proved by comparing the logistic regression model's benchmark accuracy on the original dataset with the obtained accuracy on synthetic dataset. Table 1. Whereas in the case of statistical specific metric, the results were not promising which suggest that the generators were not able to replicate the normal and empirical distributions. Table 2. As for the analysis specific metrics, the experiment was limited to only one dataset because this metric test was conducted just to have a glimpse whether the generated synthetic data can be used as a supplement or not. As per the comparison of figure 1 &2 with figures 3 , 4, 5, 6, 7 , & 8 it can be concluded that the synthetic version is not ready to be used as replacement for original dataset when it comes to exploratory data analysis. In conclusion, this experiment has proved that producing a perfect synthetic dataset is still work in progress. Further work needs to be done in this field. This experiment was only focused on creating synthetic version of a tabular dataset which means that to work with a dataset that is not following the format of a table (such as images or raw sentences), we have to find some alternate techniques or algorithms that can model the similarity between these sets. Apart from working with non-tabular datasets, another filed in which further experiment can be conducted is injecting the generated synthetic points in the original dataset to handle class imbalance. In this study we have talked about SMOTE which is executing the same idea, but this arena can be explored further where we can insert the synthetic version as much as we want. Apart from exploring new ideas, in future we also have to improve the existing problems which is improvement of the pre-developed models so that they can replicate the same statistical structure as the original dataset. This achievement can not only reduce our requirement for data in machine learning or artificial intelligence related task but can also help in using a substitute for conducting exploratory data analysis.

6. References

[1] Dilmegani, C. (2018). "SYNTHETIC DATA: in-Depth Synthetic Data Guide: What is it?How does it enable AI?" [Online]. Available: -- https://research.aimultiple.com/synthetic-data/#:~:text=%20There%20are%20several%20additional%20benefits%20to%20using,for%20data%20that%20contains%20sensitive%20information%20More%20. [Accessed on 18-05-2022].

[2] Supriya (2018). "How Big Data Plays an Important Role in Our Daily Life." [Online]. Available: https://www.upgrad.com/blog/how-big-data-plays-an-important-role-in-our-daily-life/#:~:text=%20Let%E2%80%99s%20see%20how%20Big%20Data%20impacts%20our,Banks%20can%20efficiently%20track%20and%20trace...%20More%20. [Accessed on 31-05-2022].

[3] Barbierato, E., et al. (2022). "A Methodology for Controlling Bias and Fairness in Synthetic Data Generation." Applied Sciences **12**(9): 4619.

[4] Watts, J. (2021). "AI and the importance of data management." [Online]. Available: https://www.techradar.com/news/ai-and-the-importance-of-data-management. [Accessed on 31-05-2022].

[5] telusinternational.com (2021). "Seven types of data bias in machine learning." [Online]. Available: https://www.telusinternational.com/articles/7-types-of-data-bias-in-machine-learning#:~:text=Data%20bias%20in%20machine%20learning%20is%20a%20type,skewed%20outcomes%2C%20low%20accuracy%20levels%2C%20and%20analytical%20errors. [Accessed on 02-06-2022].

[6] Fogg, S. (2019). "GDPR for Dummies: Simple GDPR Guide for Beginners." [Online]. Available: https://termly.io/resouces/articles/gdpr-for-dummies/. [Accessed on 02-06-2022].

[7] geeksforgeeks.com (2022). "Understanding Logistic Regression." [Online]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression/#:~:text=Logistic%20regression%20is%20basically%20a%20supervised%20classification%20algorithm.,popular%20belief%2C%20logistic%20regression%20IS%20a%20regression%20model. [Accessed on 02-06-2022].

[8] technopedia.com (2022). "Synthetic Data." [Online]. Available: https://www.techopedia.com/definition/33305/synthetic-data#:~:text=Synthetic%20data%20is%20created%20without%20actual%20driving%20organic,that%20would%20be%20an%20example%20of%20synthetic%20da. [Accessed on 02-06-2022].

[9] sdv.dev (2022). "Gaussian Copula Model." from https://sdv.dev/SDV/user_guides/single_table/gaussian_copula.html. [Accessed on 06-06-2022].

[10] Peterson, R. (2022). "Entity Relationship (ER) Diagram Model with DBMS Example." https://www.guru99.com/er-diagram-tutorial-dbms.html. [Accessed on 06-06-2022].

[11] sdv.dev (2022). "HMA1 Class." from https://sdv.dev/SDV/user_guides/relational/hma1.html. [Accessed on 11-06-2022].

[12] nm.dev (2022). "Radial Basis Function Neural Network." https://nm.dev/courses/introduction-to-data-science/lessons/artificial-neural-network/topic/radial-basis-function-neural-network/#:~:text=What%20is%20Radial%20Basis%20Function%20Neural%20(Radial%20Basis%20Function%20Networks)%20%20Network%3F%20The,universal%20approximation%20than%20other%20types%20of%20neural%20networks. [Accessed on 11-06-2022].

[13] wikipedia.org (2022). "Families of copulas." https://en.wikipedia.org/wiki/Copula_(probability_theory). [Accessed on 08-05-2022].

[14] Patki, N.; Wedge, R.; Veeramachaneni, K. The Synthetic Data Vault. In Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, 17–19 October 2016; pp. 399–410.

[15] 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.

[16] Nowok, B. Utility of Synthetic Microdata Generated Using Tree-Based Methods. UNECE Stat. Data Confidentiality Work Sess. 2015. Available online: https://unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/20150/Paper_33_Session_2_-_Univ._Edinburgh__Nowok_.pdf [Accessed on 26 February 2021]

[17] Brownlee, J. (2019). "A Gentle Introduction to Imbalanced Classification." [Online]. Available: https://machinelearningmastery.com/what-is-imbalanced-classification/. [Accessed on 09-06-2022].

[18] PrivBayes: Private Data Release via Bayesian Networks: ACM Transactions on Database Systems: Vol 42, No 4. Available online: https://dl.acm.org/doi/10.1145/3134428 [Accessed on 24 - 12 - 2020].

[19]. corporatefinanceinstitute.com (2022). "Cumulative Frequency Distribution." https://corporatefinanceinstitute.com/resources/knowledge/other/cumulative-frequency-distribution/. [Accessed on 09-06-2022].

[20] Practical Synthetic Data Generation [Book]. Available online: https://www.oreilly.com/library/view/practical-synthetic-data/ 9781492072737/ [Accessed on 06 - 09 - 2020]

[22] Woo, M.-J.; Reiter, J.P.; Oganian, A.; Karr, A.F. Global Measures of Data Utility for Microdata Masked for Disclosure Limitation. J. Priv. Confid. 2009, 1.

[23] Xu, L., et al. (2019). "Modeling tabular data using conditional gan." Advances in Neural Information Processing Systems **32**.

[24] Dalianis H. Evaluation metrics and evaluation. In: Clinical Text Mining. Cham: Springer; 2018:45-53.

[25] sdv.dev (2022). "Evaluation Framework." from https://sdv.dev/SDV/user_guides/evaluation/evaluation_framework.html. [Accessed on 09-06-2022].

[26] HAYES, A. (2022). "Null Hypothesis." https://www.investopedia.com/terms/n/null_hypothesis.asp. [Accessed on 09-07-2022].

[27] sdv.dev (2022). "What is Coupa Gan?". from https://sdv.dev/SDV/user_guides/single_table/copulagan.html#what-is-copulagan. [Accessed on 09-07-2022].

[28] openml.org (2022). "32 Sick Dataset." https://www.openml.org/search?type=data&sort=runs&id=38&status=active. [Accessed on 15-08-2022].

[29] openml.org (2022). "185 Baseball." https://www.openml.org/search?type=data&sort=runs&id=185&status=active. [Accessed on 15-08-2022].

[30] openml.org (2022). "Ozone Level." https://www.openml.org/search?type=data&sort=runs&id=1487&status=active. [Accessed on 18-08-2022].

[31] William, K. (2022). "What is SPSS? Definition, Features, Types, and Use Cases." https://surveysparrow.com/blog/what-is-spss/. [Accessed on 18-07-2022].

[32] microsoft.com (2015). "WikiQA: A Challenge Dataset for Open-Domain Question Answering." https://www.microsoft.com/en-us/research/publication/wikiqa-a-challenge-dataset-for-open-domain-question-answering/. [Accessed on 18-08-2022].

[33] Zibran, M. F. (2007). "Chi-squared test of independence." Department of Computer Science, University of Calgary, Alberta, Canada **1**(1): 1-7.

[34] Pratt, J. W. and J. D. Gibbons (1981). Kolmogorov-Smirnov two-sample tests. Concepts of nonparametric theory, Springer**:** 318-344.

[35] Dankar, F. K. and M. Ibrahim (2021). "Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation." Applied Sciences **11**(5): 2158.

[36] Meyer, D. and T. Nagler (2021). "Synthia: Multidimensional synthetic data generation in Python." Journal of Open Source Software **6**(65): 2863.

[37] M. Sklar, “Fonctions de repartition an dimensions et leurs marges,” Publ. inst. statist. univ. Paris, vol. 8, pp. 229–231, 1959.

[38] S.-C. Kao, H. K. Kim, C. Liu, X. Cui, and B. L. Bhaduri, “Dependencepreserving approach to synthesizing household characteristics,” Transportation Research Record, vo

[39] Lakshmi, T. J. and C. S. R. Prasad (2014). A study on classifying imbalanced datasets. 2014 First International Conference on Networks & Soft Computing (ICNSC2014), IEEE.

[40] Krishni (2018). "K-Fold Cross Validation." https://medium.datadriveninvestor.com/k-fold-cross-validation-6b8518070833. [Accessed on 18-07-2022].

7. Appendices

Appendix 1:

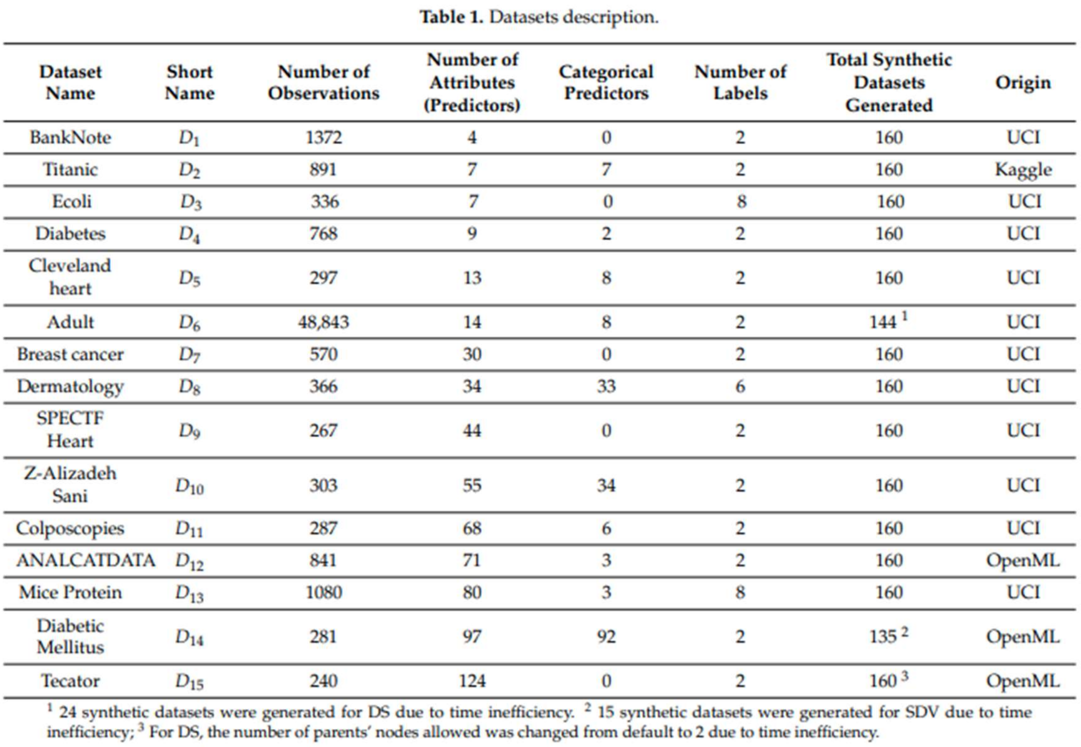
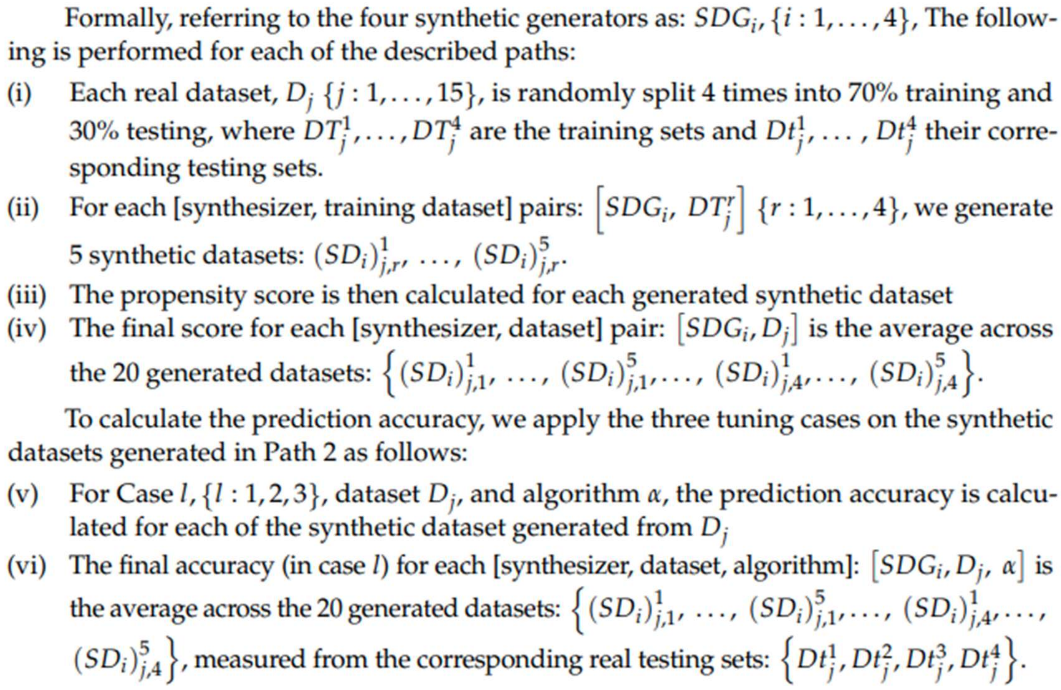


Table 1. Datasets description.

Dataset Name	Short Name	Number of Observations	Number of Attributes (Predictors)	Categorical Predictors	Number of Labels	Total Synthetic Datasets Generated	Origin
BankNote	D_1	1372	4	0	2	160	UCI
Titanic	D_2	891	7	7	2	160	Kaggle
Ecoli	D_3	336	7	0	8	160	UCI
Diabetes	D_4	768	9	2	2	160	UCI
Cleveland heart	D_5	297	13	8	2	160	UCI
Adult	D_6	48,843	14	8	2	144 [1]	UCI
Breast cancer	D_7	570	30	0	2	160	UCI
Dermatology	D_8	366	34	33	6	160	UCI
SPECTF Heart	D_9	267	44	0	2	160	UCI
Z-Alizadeh Sani	D_{10}	303	55	34	2	160	UCI
Colposcopies	D_{11}	287	68	6	2	160	UCI
ANALCATDATA	D_{12}	841	71	3	2	160	OpenML
Mice Protein	D_{13}	1080	80	3	8	160	UCI
Diabetic Mellitus	D_{14}	281	97	92	2	135 [2]	OpenML
Tecator	D_{15}	240	124	0	2	160 [3]	OpenML

[1] 24 synthetic datasets were generated for DS due to time inefficiency. [2] 15 synthetic datasets were generated for SDV due to time inefficiency; [3] For DS, the number of parents' nodes allowed was changed from default to 2 due to time inefficiency.

Appendix 2:



Formally, referring to the four synthetic generators as: $SDG_i, \{i : 1, \ldots, 4\}$, The following is performed for each of the described paths:

(i) Each real dataset, D_j $\{j : 1, \ldots, 15\}$, is randomly split 4 times into 70% training and 30% testing, where $DT_j^1, \ldots, DT_j^4$ are the training sets and $Dt_j^1, \ldots, Dt_j^4$ their corresponding testing sets.

(ii) For each [synthesizer, training dataset] pairs: $\left[SDG_i,\ DT_j^r\right]$ $\{r : 1, \ldots, 4\}$, we generate 5 synthetic datasets: $(SD_i)_{j,r}^1,\ \ldots,\ (SD_i)_{j,r}^5$.

(iii) The propensity score is then calculated for each generated synthetic dataset

(iv) The final score for each [synthesizer, dataset] pair: $\left[SDG_i, D_j\right]$ is the average across the 20 generated datasets: $\left\{(SD_i)_{j,1}^1,\ \ldots,\ (SD_i)_{j,1}^5, \ldots,\ (SD_i)_{j,4}^1, \ldots,\ (SD_i)_{j,4}^5\right\}$.

To calculate the prediction accuracy, we apply the three tuning cases on the synthetic datasets generated in Path 2 as follows:

(v) For Case l, $\{l : 1, 2, 3\}$, dataset D_j, and algorithm α, the prediction accuracy is calculated for each of the synthetic dataset generated from D_j

(vi) The final accuracy (in case l) for each [synthesizer, dataset, algorithm]: $\left[SDG_i, D_j,\ \alpha\right]$ is the average across the 20 generated datasets: $\left\{(SD_i)_{j,1}^1,\ \ldots,\ (SD_i)_{j,1}^5, \ldots,\ (SD_i)_{j,4}^1, \ldots,\ (SD_i)_{j,4}^5\right\}$, measured from the corresponding real testing sets: $\left\{Dt_j^1, Dt_j^2, Dt_j^3, Dt_j^4\right\}$.

Appendix 3:

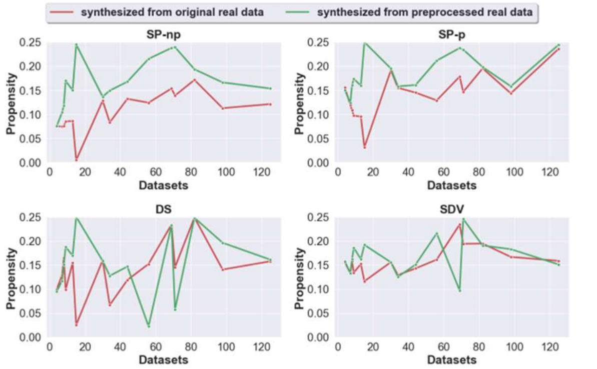


Figure 3. Propensity scores per dataset across all classification algorithms. The x-axis represents the different datasets ordered according to their attributes number d.

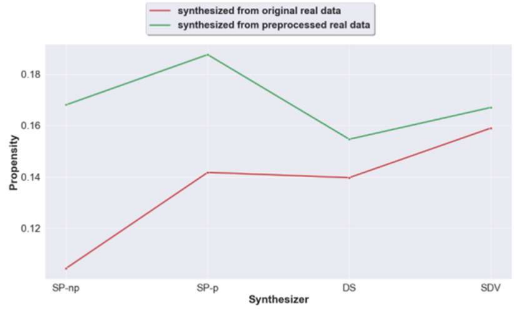


Figure 4. Propensity scores per synthesizer (across datasets).

Appendix 4:

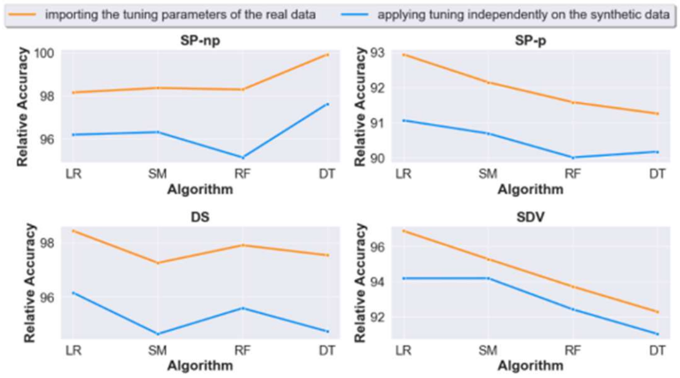


Figure 5. Relative accuracy for each synthesizer across all datasets for cases 1 and 2. The *x*-axis represents the ML model.

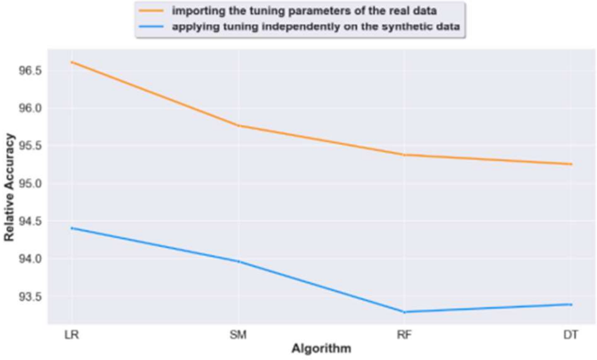


Figure 6. Relative accuracy across all synthesizers and all datasets for each ML model.

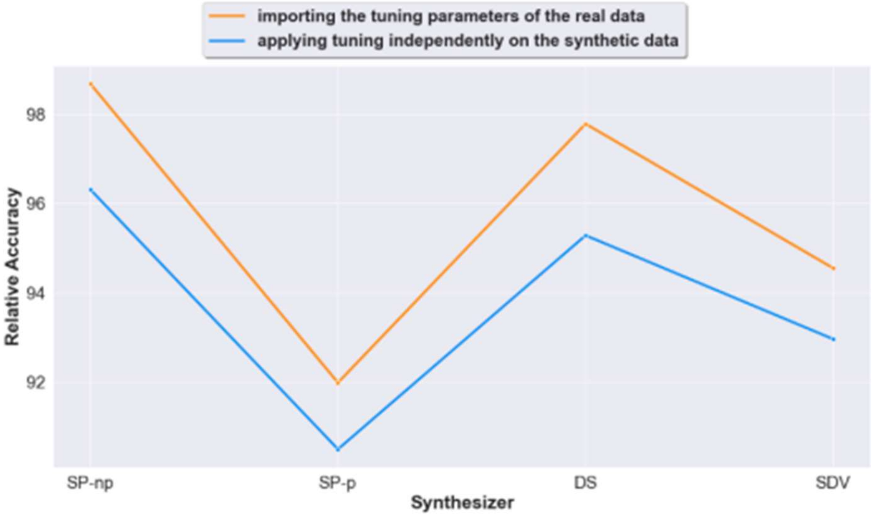


Figure 7. Relative accuracy across all ML models and all datasets for each synthesizer.

Appendix 5:

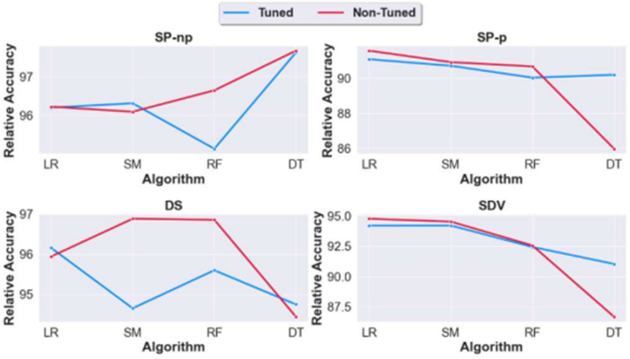


Figure 8. Relative accuracy of the ML models for each synthesizer across all datasets. Tuned and non-tuned case.

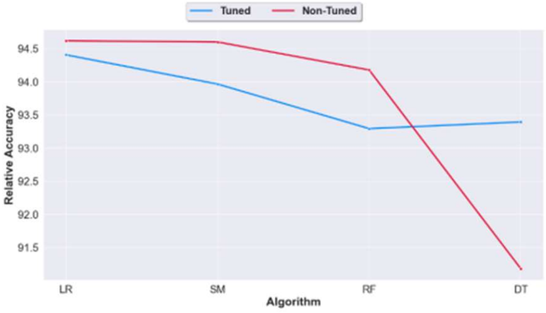


Figure 9. Relative accuracy of the ML models across all synthesizer and datasets. Tuned and non-tuned case.

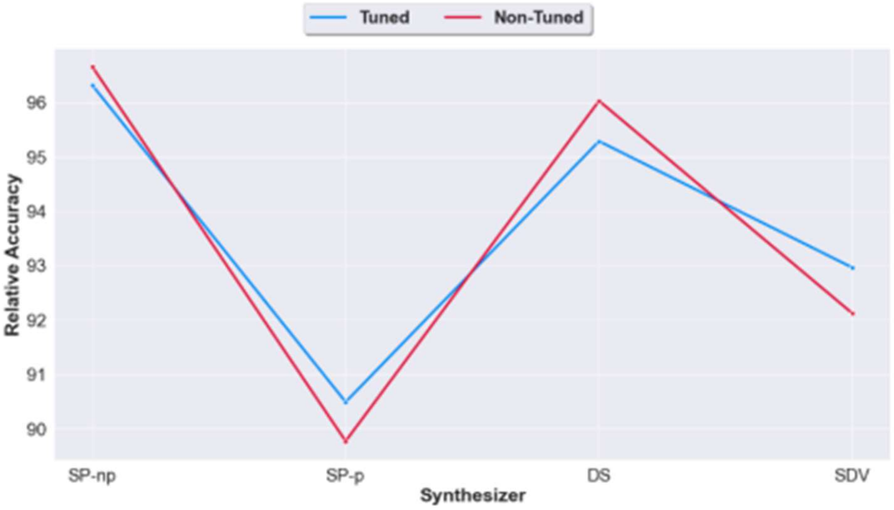


Figure 10. Relative accuracy of the synthesizers across all ML models and datasets. Tuned and non-tuned case.

Appendix 6:

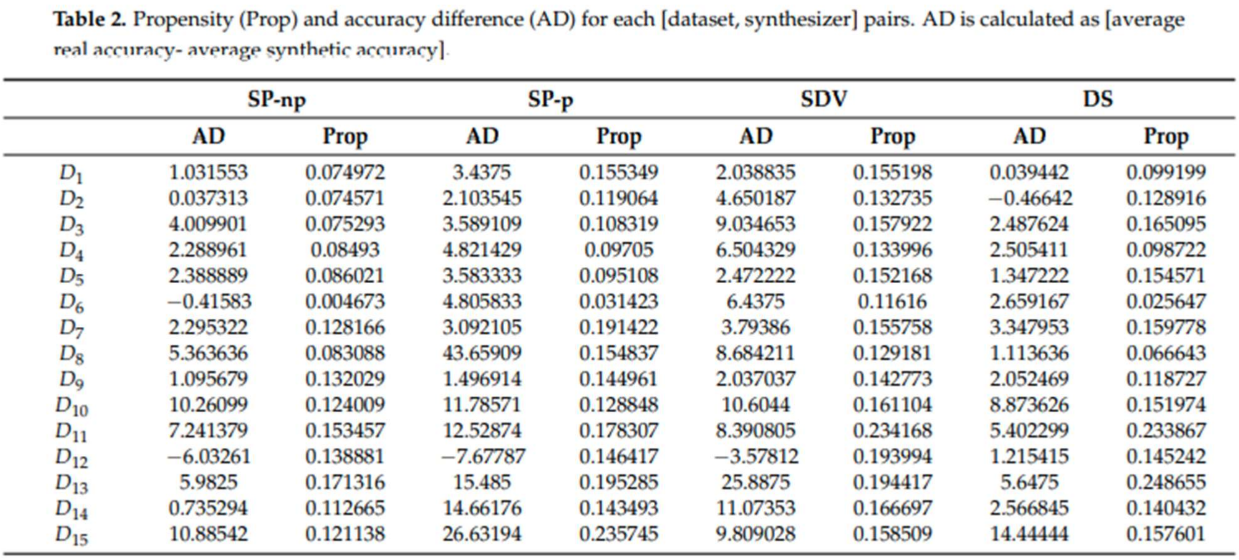


Table 2. Propensity (Prop) and accuracy difference (AD) for each [dataset, synthesizer] pairs. AD is calculated as [average real accuracy- average synthetic accuracy].

	SP-np		SP-p		SDV		DS	
	AD	Prop	AD	Prop	AD	Prop	AD	Prop
D_1	1.031553	0.074972	3.4375	0.155349	2.038835	0.155198	0.039442	0.099199
D_2	0.037313	0.074571	2.103545	0.119064	4.650187	0.132735	−0.46642	0.128916
D_3	4.009901	0.075293	3.589109	0.108319	9.034653	0.157922	2.487624	0.165095
D_4	2.288961	0.08493	4.821429	0.09705	6.504329	0.133996	2.505411	0.098722
D_5	2.388889	0.086021	3.583333	0.095108	2.472222	0.152168	1.347222	0.154571
D_6	−0.41583	0.004673	4.805833	0.031423	6.4375	0.11616	2.659167	0.025647
D_7	2.295322	0.128166	3.092105	0.191422	3.79386	0.155758	3.347953	0.159778
D_8	5.363636	0.083088	43.65909	0.154837	8.684211	0.129181	1.113636	0.066643
D_9	1.095679	0.132029	1.496914	0.144961	2.037037	0.142773	2.052469	0.118727
D_{10}	10.26099	0.124009	11.78571	0.128848	10.6044	0.161104	8.873626	0.151974
D_{11}	7.241379	0.153457	12.52874	0.178307	8.390805	0.234168	5.402299	0.233867
D_{12}	−6.03261	0.138881	−7.67787	0.146417	−3.57812	0.193994	1.215415	0.145242
D_{13}	5.9825	0.171316	15.485	0.195285	25.8875	0.194417	5.6475	0.248655
D_{14}	0.735294	0.112665	14.66176	0.143493	11.07353	0.166697	2.566845	0.140432
D_{15}	10.88542	0.121138	26.63194	0.235745	9.809028	0.158509	14.44444	0.157601

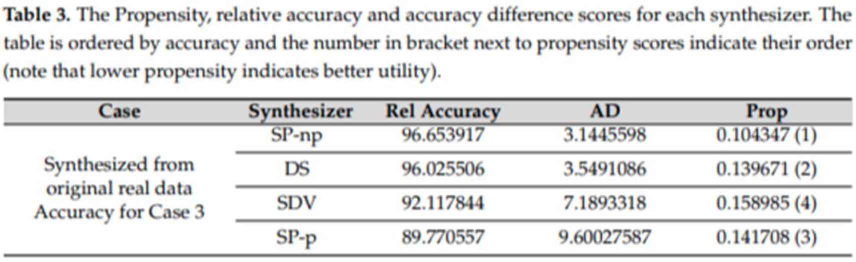


Table 3. The Propensity, relative accuracy and accuracy difference scores for each synthesizer. The table is ordered by accuracy and the number in bracket next to propensity scores indicate their order (note that lower propensity indicates better utility).

Case	Synthesizer	Rel Accuracy	AD	Prop
Synthesized from original real data Accuracy for Case 3	SP-np	96.653917	3.1445598	0.104347 (1)
	DS	96.025506	3.5491086	0.139671 (2)
	SDV	92.117844	7.1893318	0.158985 (4)
	SP-p	89.770557	9.60027587	0.141708 (3)

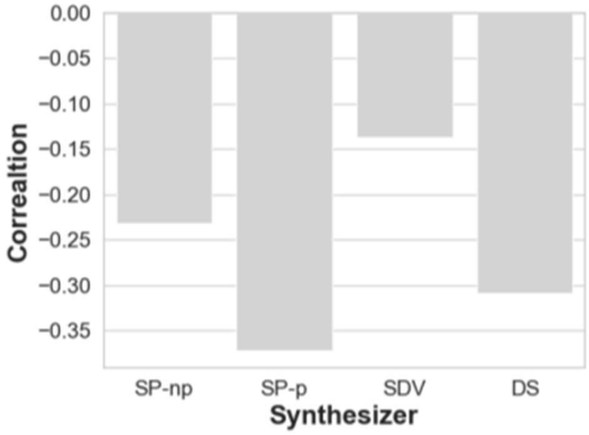
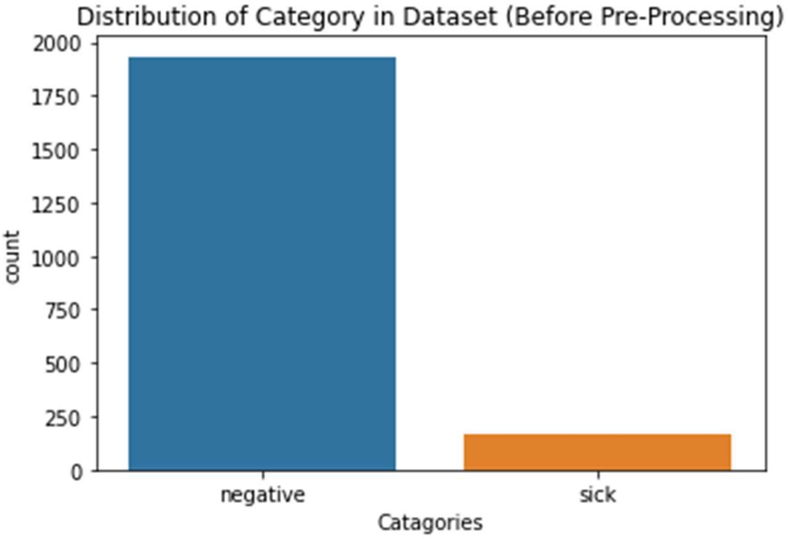
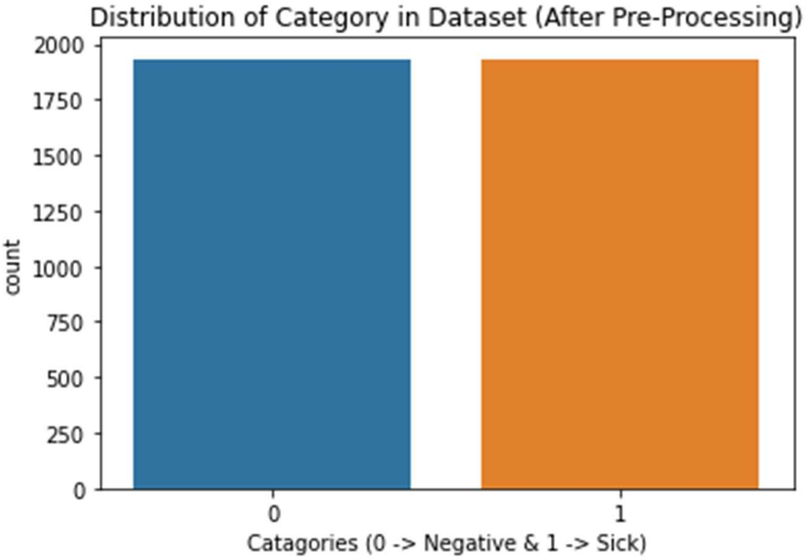


Figure 11. Correlation coefficients per synthesizer.

Appendix 7:


Distribution of Category in Dataset (Before Pre-Processing)
2000
1750
1500
1250
1000
750
500
250
0
count
negative
sick
Catagories

Appendix 8:


Distribution of Category in Dataset (After Pre-Processing)
2000
1750
1500
1250
1000
750
500
250
0
count
0
1
Catagories (0 -> Negative & 1 -> Sick)