École Polytechnique Fédérale de Lausanne

Department of Mathematics

Master in Computational Science and Engineering

Synthetic data assessment based on model improvement

Master Thesis

In partial fulfilment of the requirements for the Master of Science EPFL in Computational Science and Engineering (MSc EPFL)

Submitted by:

Romain Palazzo

Supervisor(s):

Kalyan Veeramachaneni

Principal Research Scientist, MIT

Nicolas Boumal

Tenure Track Assitant Professor, EPFL

Tutor:

Dongyu Liu

Postdoctoral Associate, MIT

Cambridge, July 2022

Abstract

Synthetic data appears to be one of the most promising solutions for solving data limitation problems in Machine Learning (ML). Thanks to computers and new models, synthetic data can now be generated quickly and in large amounts. However to ensure this data is realistic and useful, it must be assessed.

In this thesis, we investigate forms of synthetic data assessment based on model improvement. In other words, we evaluate synthetic data according to its capacity to improve ML models when it is added to the model training process. To do this, we first performed a review of the existing works and metrics for assessing synthetic data. Then we proposed new metrics based on unexploited statistical and ML notions. We finally built regression models that are able to predict model improvement from these metrics' scores. Those models have a R^2 coefficient 0.72 over a cross-validation and a correlation between the ground truth and the prediction of 0.8.

Contents

List of Figures

List of Tables

Symbol	Description
$\mathscr{D}_R$	Real dataset
$\mathscr{D}_s$	Synthetic dataset
$\mathscr{D}_i$	Dataset i
$f_a(x)$	New improvement function
$S_\alpha(p)$	$p-value$ re-mapping
n	Number of samples real data
m	Number of samples synthetic data
s_i	Metric i
ϕ_R	Friedman-Rafsky statistics
ϕ_E	Energy Test statistics
ϕ_k	KNN Test statistics
ϕ_{MMD}	MMD Test statistics
M_{SHAP}	Shapley value metric 1
O_{SHAP}	Shapley value metric 2
M_{FAMD}	FAMD based metric

Table 1: Notation Table

List of Acronyms

LD Logistic Detection 20

MLP MultiLayer Perceptron 7

SVCD SVC Detection 20

BDTC Binary Decision Tree Classifier 20

BABC Binary Ada Boost Classifier 20

BLR Binary Logistic Regression 20

BMLPC Binary MultiLayer Perceptron (MLP) Classifier 20

MCDTC Multi-class Decision Tree Classifier 20

MMLPC Multi-class MLP Classifier 20

LR Linear Regression 20

MLPR MLP Regressor 20

GMLL GM Log-Likelihood 19

CSTest Chi-Square Test 18

KSTest Kolmogorov-Smirnov Test 18

PCA Principal Components Analysis 2

Acknowledgments

This master's thesis has been a short, intense and beautiful dive into the world of research and I would like to thank the people who supported me in this fantastic experience.

First and foremost, I would like to thank Kalyan Veeramachaneni, who trusted and helped me from the first day, way before the trip to Boston. He made one of my dreams come true, of joining a MIT laboratory for my master's thesis. I'm also grateful for all the guidance and expertise he gave me. Kalyan is an amazing advisor who truly cares about of all his students.

Then, I would like to thank Dongyu Liu for having supervised my thesis all along. Our discussions as well as his regular feedback and scientific knowledge have greatly contributed to my work and helped me to build up my scientific and critical thinking.

I would also like to thank my labmates in the DAI lab. I met a team of amazing people, excellent in their work but even more importantly demonstrating a rare kindness. The ski trip or the retreat are among my best memories in Boston. It was a pleasure to come to the office every day with you.

I feel blessed and grateful for everyone who has helped me finish my thesis. Especially I'd like to thank Laure Berti-Equille for giving me precious advice concerning my last results and how to present them.

Special thanks to Nicolas Boumal for having agreed to supervise this project.

Lastly, I would like to thank my loving family for their continuing love and support, even with the distance – especially my parents Murielle and Fabrice and my brother Nicolas, who have always been my driving examples.

To my parents Murielle and Fabrice, and my brother Nicolas.

1 Introduction

Statistics as a field uses data to derive knowledge. Machine Learning (ML) has become a powerful and increasingly popular statistical tool. The two main subfields of ML are **supervised and unsupervised learning**. Supervised learning involves training models on *labeled* data and using them to predict outcomes (usually defined as "*labels*". In this thesis, we are only interested in supervised learning, especially **classification tasks**. Compared to a **regression task**, where the *label* is a continuous variable, in a classification task the *label*s are categories. Fig.1 shows the workflow of how a ML model is built and evaluated for a classification task. The data is first split into **train and test sets**, and the model learns to predict the label from the train set. Then, model performance is assessed by computing the score over the test set.

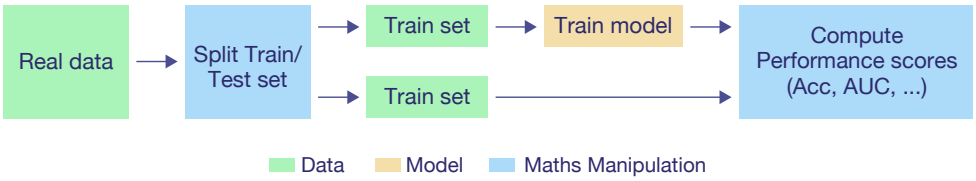

Figure 1: Training/Testing procedure

One of the common problems machine learning model developers face is lack of labeled data. These situations are much too common place in today's usage of machine learning. In these situations, one of the common approach is to possibly augment synthetic data to training data. In this thesis, we use synthetic data as a **data augmentation** method – data added to train the model. In Section 1.1 we introduce and define synthetic data and give the motivation of using synthetic data for machine learning in Section 1.2.

1.1 What is synthetic data?

According to the McGraw-Hill Dictionary of Scientific and Technical Terms[1], synthetic data is " any production data applicable to a given situation that is not obtained by direct measurement" Synthetic data has no real origin, but its systematic and large-scale use is characteristic of our time [2]. This is a result of the growing need for data and the ease of its creation thanks to the computational methods now available to generate such data.

Machine Learning created the need for synthetic data, and solved it by proposing efficient synthetic data generators. Indeed, Machine Learning models are now used across society, and all of them need data to learn and to be trained. However, the larger the architecture of an ML model, the more data is required to produce viable results. To fulfill this need for data, new ML models have been created to produce large amounts of high-quality synthetic data [3]. "High-quality" in this case means that the data is hard to distinguish from real data, even for a human. Generative Adversarial Network (GAN) is one of the most popular techniques for synthetic data generation.

There are many games where one has to distinguish real images from images generated by GANs. [4]

Synthetic data has multiple applications, primarily **data augmentation** – when one wants more data from an experiment without repeating it. This can serve a number of purposes, from model improvement to statistical data analysis.
The second application is **data privacy** [5]. In multiple domains, such as healthcare, it's helpful to be able to work with data (for instance, to clean or process it) without sharing sensitive information. In this situation, the cleaning and processing method can be developed and tested over synthetic data and then used with the sensitive data. A third application is **population or experiment synthesis**, where data is created to represent situations that don't exist or experiments that cannot be done yet.

1.2 Use of synthetic data for data augmentation : Concrete example

We present the following example to explain why synthetic data is useful as a data augmentation method. Let's consider the dataset `vineyard`, which has shape (52,3): fifty-two rows and three columns. The column `class` is the binary target column, and `lugs_1989`, `lugs_1990` are two continuous columns (*a.k.a* feature). We then created a gaussian copula model to fit the data and generate a synthetic dataset. (This synthetic generator is not a GANs and will be explained further in the Appendix.) Thanks to this model, we can generate synthetic data with the shape (52,3). We trained classical ML classifiers, first with only real data in the training and test set, and then with increasing amounts of synthetic data added to the training set. The test set always stayed the same, composed only of real data. We computed classical model performance scores after each synthetic data addition; i.e., the Accuracy, Recall, AUC, F1, and Prec. All performance scores are defined in the Appendix, Section7.1. A diagram of this experiment is given in Fig.2.

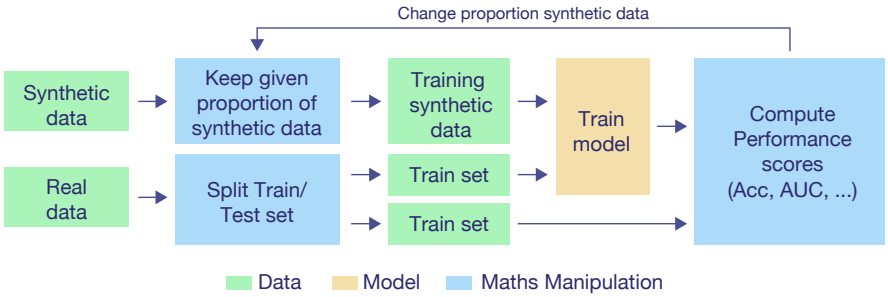

Figure 2: A data augmentation experiment with a simple 3 columns dataset

Figure 3 shows how we track changes in model performance when synthetic data is added. Score values range between 0 and 1, with 1 corresponding to the best possible score. In the case of Accuracy, for instance, a score of 1 means that all the model predictions over the test set are correct. In Fig.3. The $x-axis$ is the proportion of synthetic data in the training set, and the $y-axis$ is the performance score. When there is no synthetic data in the training set, the model performances are not bad but can be improved. The F1 score, AUC, and Precision metrics perform especially poorly, with scores under 0.5. As is clear from the chart, adding synthetic data greatly improves the model performance. Finally, when there is as much synthetic as real data in the training set, four out of the five model performance metrics score above 0.8, which is good. This example helps to show that synthetic data can be used as a data augmentation method to improve model performance.

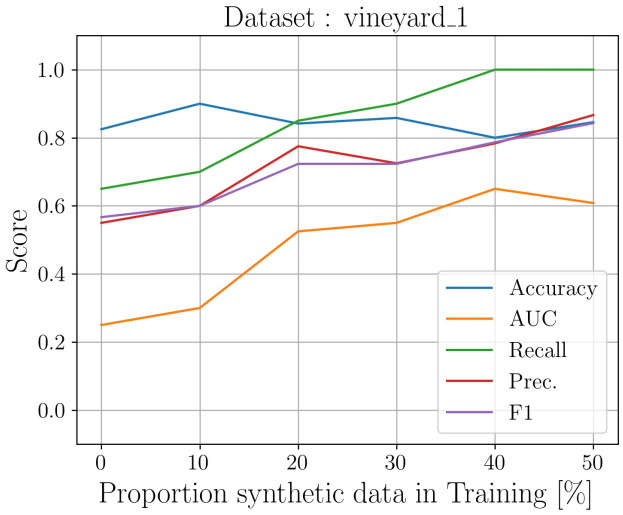

Figure 3: Model performance improvement thanks to synthetic data addition in the Training set

1.3 Case study and challenges

Can we predict whether data augmentation with a synthetic dataset will improve a model? This is the central question of this thesis. The motivations behind this question include **saving time** and **improving synthetic data generation**. Indeed, because training is the most time-consuming part of building an ML model, we would like to know whether a synthetic dataset will improve a model without re-training. If we're able to predict the amount improvement, it becomes possible to build a synthetic data generator that uses this information to create high-quality synthetic data.

To answer this central question, we work with the following setup. First, we consider only **tabular data** built to perform a classification task. To generalize our results, we consider multiple datasets from various fields (science, travel, and economics). For each dataset, we compute metrics designed for data assessment, and from those metrics, we build models to predict improvement. The output of those models comprises our evaluation of synthetic data. A good score means that the ML model should improve with data augmentation and a bad score suggests the opposite outcome.

Challenges. Before delving deeper into the thesis, it's important to have the following in mind. In Machine Learning, data is almost everything, but not quite. That is to say, even if you have the most realistic synthetic data, you may not improve your model performance by adding it to the training, because of model and experiment limitations.
A **model limitation** may occur when a model doesn't have a good structure or learning process for extracting information from the data. An exaggerated but easily understandable example is that of training a regression model to perform a classification task. The output of a regression model is

continuous, while classification expects categories as output. Clearly, this poorly designed structure will limit the efficacy of the model.
An **experiment limitation** in our case is when the target column is difficult to predict from the others because there are no clear correlations between them. Knowing that, we will try to propose the best metrics and aggregate them in the most useful way for a user. A good final score should suggest that model performance could be improved by adding the given synthetic dataset into the training.

1.4 thesis overview

To end this introduction, we give more details about the structure of this thesis. This thesis is divided into 4 main steps, corresponding to the colored rectangles in Fig.4. Fig.4 gives a quick description of every step, as well as their outcomes.

Dataset Pre-Processing : In this step we generate synthetic data for the all datasets we have, and we select the ones best suited to our experiments. Section.2.1-2.2

SDMetrics Review : Here we review existing metrics that assess synthetic data quality. The goal is to analyze current strategies, and examine their relevance and implementation. Section.3.3.1, and 3.4 for the results.

New metrics proposal : After this review, we propose new metrics for assessing synthetic data quality. Those new metrics are based on unexploited statistical and Machine Learning fields. Section.3.3.3, 3.5

Optimization problem : In the final part of the thesis, we use existing and new metrics to build ML models that are able to predict model improvement. Section.4

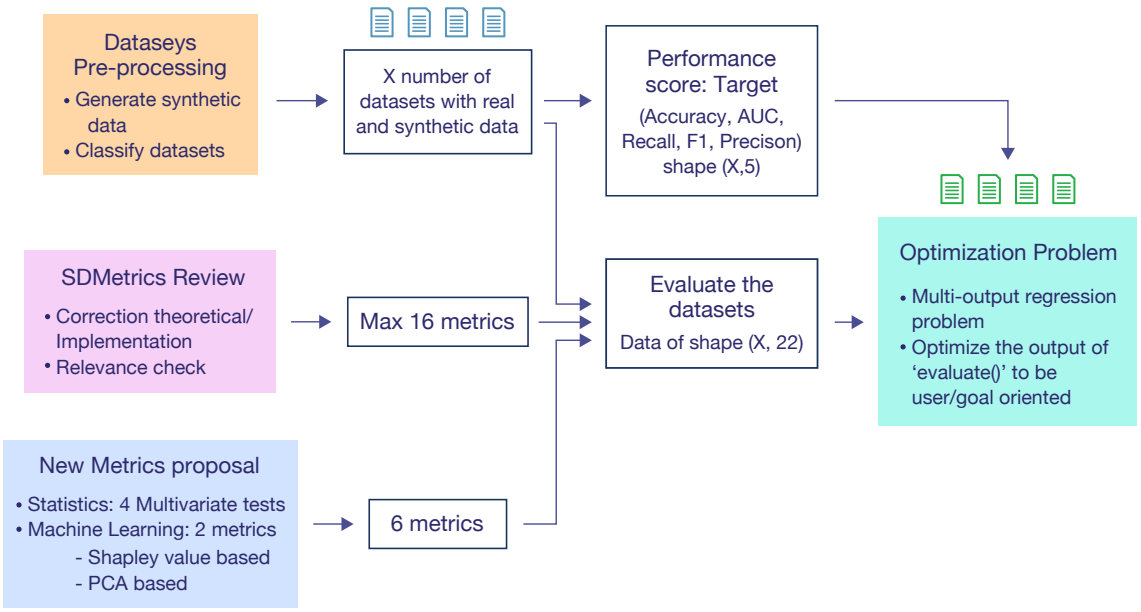

Figure 4: Overview of the thesis

2 Preliminaries

2.1 Datasets description

We use of a large number of datasets in this thesis. We begin with 466 different tabular datasets. Tab.2 presents some statistics about these datasets in order to illustrate their diversity. While these datasets come from different fields, such as biology, physics or healthcare, all contain a target column with classes and all have been built to solve a classification task. Because we want to use synthetic data for data augmentation, we don't want to work with datasets that already lead to models that perfectly or almost perfectly predict the target. Therefore, this first step of the thesis is generating synthetic data for all the datasets, and selecting those datasets that can gain from data augmentation.

	Mean	**Median**	**Min**	**Max**
Number of rows	4623	500	20	245057
Number of columns	127	11	2	10936
Number of continuous columns	99	7	0	10935
Number of categorical columns	28	2	1	4703
Number of class	2.32	2	2	10

Table 2: Statistics concerning datasets

2.2 Datasets selection

The 466 datasets used in this thesis can all be used for classification tasks. However, dealing with this amount of data is challenging. For this reason, and because some datasets are not suited for our thesis, we perform a **dataset selection**.

All datasets for which adding synthetic data in the training is useless are not suitable for our experiment. This happens in two main situations: First, when a high-performing classification model can be built using only real data. In this case, synthetic data is not needed for data augmentation, because the model is already good enough. The other situation is when the classification task is too hard to be solved by a classical classifier. We detect this when the performance scores are low when the model is trained over real data. Even if the synthetic data is close to real data, adding it to the training will not improve model performance because the task is too hard.

In Fig.5, the $x-axis$ is the performance score achieved when the model is trained only on real data, called the score baseline. In this figure, one can see that around 30 datasets already reach performances of 1 in almost all the performance metrics considered: in other words, performance improvement is almost impossible. Thus, we define our model selection rule based on the **Sum of Model Performance score SMP**, defined for dataset i as $\text{SMP}_i = \text{Accuracy}_i + \text{AUC}_i + \text{Recall}_i + \text{Precision}_i + \text{F1}_i$. This quantity is defined between 0 and 5. We computed it for all datasets where

it was possible to train models, with the results presented in Fig.6. We defined the datasets that are well suited for our thesis as those having an SMP such that 2<SMP<4. We did this because an SMP of four or higher indicates the average of all the model performance scores is at or above 0.8, meaning the model is good enough without adding synthetic data. Meanwhile, an SMP below two indicates the average of all the performance scores is below 0.4, which shows that the task is too hard for the models built. After filtering out the datasets that did not have SMP scores between 2 and 4, we were left with 152 datasets.

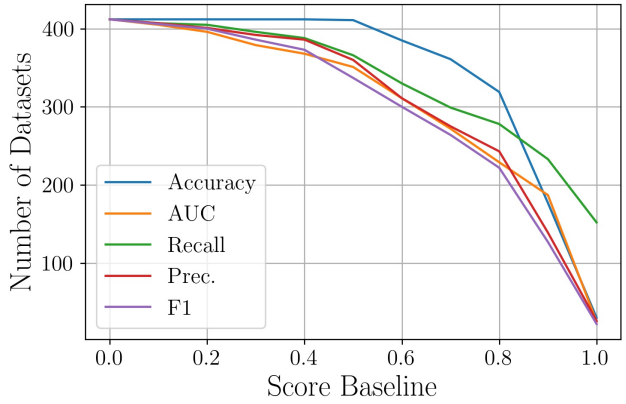

Figure 5: Profiling dataset

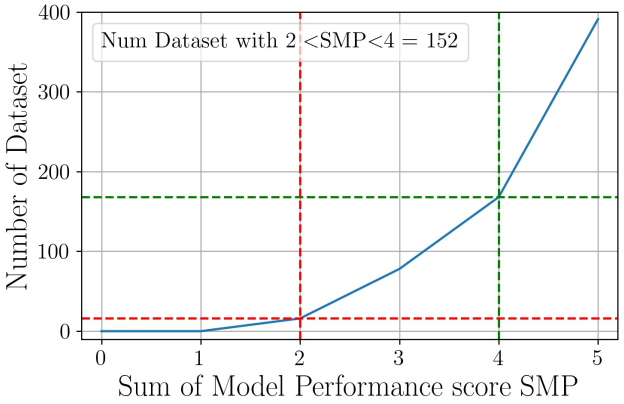

Figure 6: Dataset selection

Knowing which datasets we will work with, we are now able to generate synthetic data for them. A description of the data generation process is given just after.

2.3 Synthetic Data Generation

We have seen that adding synthetic data to a train set can improve model performances on a real test set (Section. 1.2). This was the first motivation for using synthetic data. The second one is the constant need for data. Indeed, the more complex a task, the more complex the machine learning models are, the more parameters they contain, and the more data they require. This need for data has helped synthetic data generation to become a thriving and urgent research field, where recent advances in deep learning and data generation have enabled new approaches.

In this thesis, we use one of these approaches, Generative Adversial Networks (GANs). Put simply, the GAN works by training two neural networks against each other: a generator and a discriminator. The generator processes random noise to produce synthetic data. The discriminator is a classifier trained to distinguish real from synthetic data. When they are trained against one another, the generator can create realistic data after a few iterations. While the primary application of GANs has been in generating image data, with a particular focus on human faces (Alqahtani et al., 2021), researchers have also developed specific architectures for tabular data. For example, TableGAN [6] and TGAN [7] are two specific GAN models used for tabular data. In this thesis, we used TGAN as synthetic data generator, as well as Gaussian Copula. A description of how Gaussian Copula works

is given in Appendix (Section.7.2).

2.4 How much synthetic data to add?

Usually, synthetic data is added to the train set in proportion to the real data, such that half of the samples in the train set are synthetic. We wanted to check that this was the right approach for our experiment. To do this, we compared the model improvement when different proportions of synthetic data were added to the train set.

We run this experiment and all the others over the selected datasets. For each synthetic data proportion in the train set, we compute the proportion of models improved, which is the ratio between the number of datasets where the model performances have increased with synthetic data addition over the total number of datasets. This is done for every model performance score, and the results are presented in Fig.7. From this graph, we can conclude that 0.5 is a good proportion of synthetic data to add to the train set. Indeed, considering all model performances, this results in the most model improvement. Therefore, we add the synthetic data in this proportion for the next experiment.

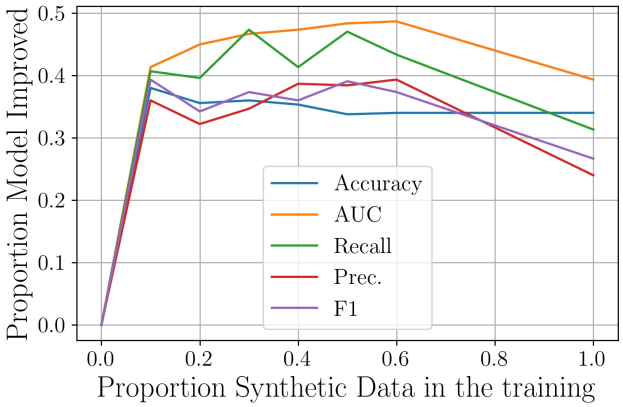

Figure 7: Grid search to find the best proportion of synthetic data in the training to improve model

Also from Fig.7, one can see that at 0.5 ($x-axis$), between 30 and 50 % of datasets have an improved model thanks to synthetic data augmentation. Also, for some datasets, models perform better when trained only with synthetic data (when the proportion of synthetic data in the training is 1).

3 Synthetic Data Assessment

As we have seen, current synthetc data generators allow us to generate thousands of data without experiment limitations. However, **how confident should we be in this data? How realistic are they? When should we use them?** Data Assessment tries to answer all these questions by evaluating data quality. To do this, the "real data" coming from an experiment is considered as ground truth, and evaluations are arrived at by comparing the properties of synthetic and real data.

In addition to synthetic data generation, The Synthetic Data Vault (SDV) [8] proposed a state-of-the-art Python library to assess synthetic data quality, called *SDMetrics*. Their end-to-end pipeline computes many metrics to evaluate synthetic data quality and aggregates their results to give a final score. A description of the metrics they use is given below.

3.1 Synthetic data assessment : what does exist?

3.1.1 Statistical metrics

These metrics are based on statistical test or probability distribution properties. For the statistical test, they corresponds to a two-sample test. A two-sample test is based around the null hypothesis $\mathscr{H}_0 : F(\mathscr{D}_r) = F(\mathscr{D}_s)$ with $\mathscr{D}_r, \mathscr{D}_s$ as real and synthetic samples respectively and F as their Cumulative Distribution Function (CDF). This means that real and synthetic data come from the same distribution under the null hypothesis. The tests implemented are the Chi-Square Test (CSTest) and Kolmogorov-Smirnov Test (KSTest).

3.1.1.1 Chi-square test. This test is defined for discrete columns. Let us consider one discrete column composed of $m+1$ categories; the statistic Q_m is given in Eq.1. Here, M_j is the number of samples of the category j in the synthetic data (m samples in total), n is the number of samples in the real data and p_j is the probability of drawing the category j. Since we only have access to the empirical distribution, we have $p_j = \frac{n_j}{n}$ with n_j as the number of samples of the category j in the real data. Then, under the null hypothesis : $\mathscr{H}_0 : \frac{M_j}{m} = \frac{n_j}{n} \quad \forall j$, Q_m follows asymptotically a χ^2 distribution with m degrees of freedom. The test is performed over all discrete columns of the tables. For each, the associated $p - value = P(X > Q_m)$ with $X \sim \chi^2(m)$ is computed, and the final metric score is the average of all the previous ones. Therefore, the metric score is included in [0,1], with 1 indicating the best possible score.

$$Q_m = \sum_{j=1}^{m+1} \frac{(M_j - np_j)^2}{np_j} \xrightarrow{d} \chi^2(m) \tag{1}$$

3.1.1.2 Kolmogorov-Smirnov test. For continuous columns, Nečasová & Svoboda in [9] used the Kolmogorov-Smirnov test to evaluate the similarity between distributions. For a sample of size k, the Empirical Cumulative Distribution Function (ECDF) is defined by $\hat{F}(x) = \frac{1}{k}\sum_{i=0}^{k} \mathbf{1}_{x_i<x}$ with $\mathbf{1}$ as the indicator function and x_i as the sample i of the dataset. Under the null hypothesis : $\mathscr{H}_0 : \hat{F}_{\mathscr{D}_s}(x) \rightarrow \hat{F}_{\mathscr{D}_r}(x)$ with $\hat{F}_{\mathscr{D}_r}$, $\hat{F}_{\mathscr{D}_s}$ as the ECDF of the real and synthetic data respectively. In these conditions, the statistic $\sqrt{m}D_m$ asymptotically follow a Kolmogorov law given by :

$$D_m = \sup_x |\hat{F}_{\mathscr{D}_r}(x) - \hat{F}_{\mathscr{D}_s}(x)| \quad \sqrt{m}D_m \xrightarrow{d} K$$

$$p - value = \mathbb{P}(X > \sqrt{m}D_m) \quad X \sim K$$

$$F_K(x) = \mathbb{P}(X \leq x) = \Big(1 + 2\sum_{j=1}^{\infty}(-1)^j e^{-2j^2x^2}\Big)\mathbf{1}_{x>0}$$

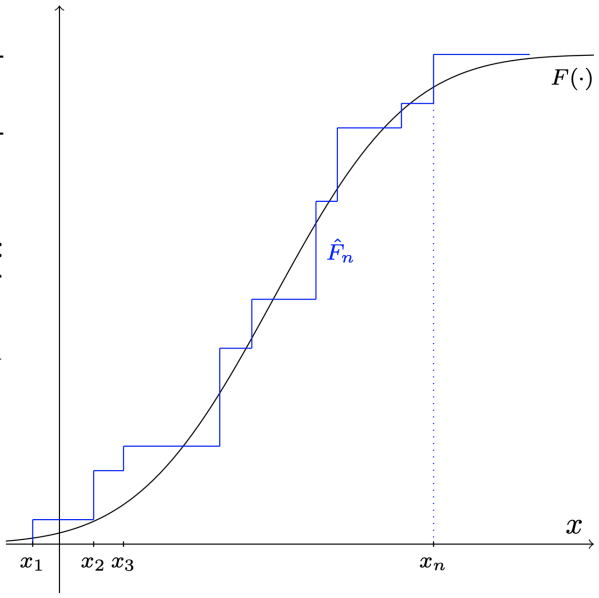

Figure 8: Empirical and theoretical distributions

Here D_m is the maximum difference in the $y - axis$ of Fig.8 between the 2 curves for a same x value. This time, the metric used in *SDMetrics* is not the $p - value$ but the inverted statistics; that is to say, $1 - D_m$. Again this test is performed for all continuous columns, and the results are averaged into one final score.

3.1.1.3 Kulback Leibler Divergence. The two last statistical metrics implemented in SDV [8] are not from a two sample test. Instead, they are measures of dissimilarity between two probability distributions. These are based on pdf (probability density function) and are defined in the discrete or continuous case by Eq.2 :

$$\mathrm{KL}(f_1, f_2) = \sum_i f_1(i)\ln(\frac{f_1(i)}{f_2(i)}) \quad \text{or} \quad \mathrm{KL}(f_1, f_2) = \int_{-\infty}^{\infty} f_1(x)\ln(\frac{f_1(x)}{f_2(x)})dx \tag{2}$$

with f_1, f_2 as the pdfs of the column for the real and synthetic data respectively. According to Eq.2, if $f_1 = f_2 \quad \forall x$ then $\mathrm{KL}(f_1, f_2) = 0$. Therefore, the best possible score for those metrics is 0. However, those metrics are not bounded, and can take values between $[-\infty,\infty]$.

3.1.1.4 Likelihood metrics. These metrics attempt to fit a probabilistic model to the real data and, later, to evaluate the likelihood of the synthetic data on it. In particular, GM Log-Likelihood (GMLL) fits multiple Gaussian Mixture models to the real data and then estimates the average log-likelihood of the synthetic data on them.

3.1.2 Metric to assess synthetic data utility for ML

3.1.2.1 Machine Learning Efficacy Metrics. This family of metrics evaluates whether it is possible to replace the real data with synthetic data to solve a Machine Learning problem. Train on Synthetic data Test on real data (TSTR) metrics was introduced in [10] in 2018. They are computed following the steps illustrated in Fig.9. For these metrics, a ML model is trained only with synthetic data and then tested over real data. The relevant metrics are the accuracy over the test set for classification task and R^2 coefficient for regression task. These metrics names are Binary Decision Tree Classifier (BDTC), Binary Ada Boost Classifier (BABC), Binary Logistic Regression (BLR), Binary MLP Classifier (BMLPC) for binary target, Multi-class Decision Tree Classifier (MCDTC), Multi-class MLP Classifier (MMLPC) for mutli-class target and Linear Regression (LR),MLP Regressor (MLPR) for regression. Here the metric name corresponds to one type of classifier/regressor.

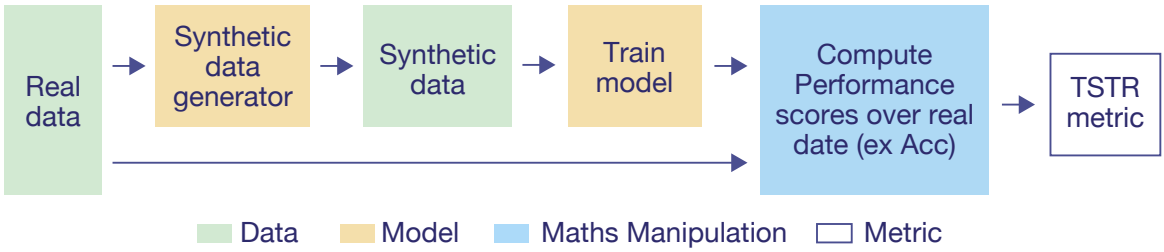

Figure 9: TSTR computation from data generation

3.1.2.2 Detection Metrics. The metrics of this family evaluate how difficult it is to distinguish the synthetic data from the real data by using a Machine Learning model. To do this, these metrics shuffle the real data and synthetic data together with flags indicating whether the data is real or synthetic, and then cross-validate a Machine Learning model that tries to predict this flag. The metric score is 1 minus the average ROC AUC score (in the sense of Section7.1.5) across all the cross-validation splits. Those metrics are Logistic Detection (LD), SVC Detection (SVCD).

3.1.2.3 Privacy metrics. To cover all considerations, *SDMetrics* also proposed privacy metrics that measure the privacy of a synthetic dataset with the following question: Can an attacker predict sensitive attributes in the real dataset trained on synthetic data? For instance, sensitive attributes might be the age or gender of an individual. Here, the metrics correspond to the Accuracy on a real test set of an attacker trained with synthetic data. Since those metrics are not as relevant for data assessment, we don't use them in this thesis.

3.2 Limitations

The previous overview of *SDMetrics* laid out their meanings and their possible values. Knowing this, some limitations can be highlighted.

3.2.1 Statistical metrics limitation : No multivariate statistical test

Concerning the statistical metrics, all the tests are performed over single columns of the dataset. Therefore, only marginal distributions are considered. However, it is easy to imagine a situation where all marginal distributions are respected, but the multivariate distribution is different. Let's take the following example :

- 2 datasets composed of 2 columns: one discrete, one continuous.
- For the discrete column, for both the column follows a Bernoulli distribution of probability $p = 0.5$, $\mathscr{B}(p)$
- For both datasets, the second column is conditionally defined according to the first.
- For the first dataset, when the value of the fist column is 1, the value of the second column is drawn from a normal law $\mathscr{N}(2,1)$. When the value of the fist column is 0, the value of the second column is drawn from a normal law $\mathscr{N}(-2,1)$
- For the second dataset, it's the opposite: When the value of the first column is 1, the value of the second column is drawn from a normal law $\mathscr{N}(-2,1)$ and when it's 0, the second column is drawn from a normal law $\mathscr{N}(2,1)$

In this sample case, performing only CSTest and KSTest as statistical tests will result in a high $p-value$ for each, while the multivariate distribution of the two datasets is different. Indeed, both marginal distributions are the same although the multivariate ones are not.

3.2.2 ML metrics limitations

ML metrics based on mixing real and synthetic data in the training. In addition, there are no Machine Learning metrics that quantifies model performances improvement when real and synthetic data are mixed in the training set, even though this is the main use for synthetic data. Indeed, section (3.1.2.1) has presented the only metrics available for assessing ML model performance, but in this case the models are trained only with synthetic data.

Redundant metrics. Finally, most of the ML *SDMetrics* do the same task, and are different only in terms of models used. For instance, BABC, BDTC, BLR and BMLPC are all binary classifiers trained with synthetic data and tested over real data. The corresponding metric is always the accuracy over the test set. This may cause redundant information if all those architectures are able to fit the data (and they are built to it). Therefore, we will try to present new ML metrics exploiting other fields of Machine Learning.

3.3 Our research goals

3.3.1 Metrics review

After the preliminaries, the first part of this thesis involves reviewing existing metrics of *SDMetrics*. Because this work focuses on data assessment, we do not consider privacy metrics. Therefore, the existing metrics studied are given in Tab.3. The key points of the review are given just below :

- **Check theoretical/Implementation of *SDMetrics*** : Analyze whether the existing metrics are theoretically good for synthetic data assessment and if the implementation is correct.
- **Assessment over real data :** Observe the score for every metric when only real data is involved. Discard those that don't give a high score.
- **Robustness analysis :** Compute the metric score variation when the same data is given multiple times. Discard those where the score varies too much.
- **Correlation study :** Compute the correlations between the *SDMetrics* to see which give similar results. This can be used to reduce the number of computed metrics.

LogisticDetection	SVCDetection	BinaryDecisionTreeClassifier	BinaryAdaBoostClassifier
BinaryLogisticRegression	MLPRegressor	MulticlassDecisionTreeClassifier	GMLogLikelihood
BinaryMLPClassifier	CSTest	MulticlassMLPClassifier	KSTestExtended
LinearRegression	KSTest	MulticlassMLPClassifier	DiscreteKLDivergence

Table 3: List of existing metrics studied

3.3.2 $p-value$ re-mapping

For now, the metric based on the CSTest and the one for the KSTest are the $p-value$ and the statistics of the test, respectively. The first change we make is to base the statistical metric only on the $p-value$ p. This is more meaningful and produces a score between 0 and 1. Moreover, metrics based on statistics do not take sample sizes into account. Having a KSTest statistic of 0.1 means different things depending on whether there are 10 or 1000 samples (in the latter situation, the p-value should be very low, suggesting a different distribution).

We also introduce the significance level α, commonly used in statistical tests, to built the metric $S_\alpha(p)$. The underlying idea behind this metric is that if one is not able to reject the null hypothesis $\mathscr{H}_0$ at the level α, $S_\alpha(p)$ should be high. Based on the criteria $S_\alpha(\alpha) = 0.5$, we defined $S_\alpha(p)$ as given in Eq.3 and visible in Fig.10 for different α. Notably, this metric is concave if $\alpha < 0.5$.

$$S_\alpha(p) = \frac{\ln(mp+1)}{\ln(m+1)} \quad m = \frac{1-2\alpha}{\alpha^2} \tag{3}$$

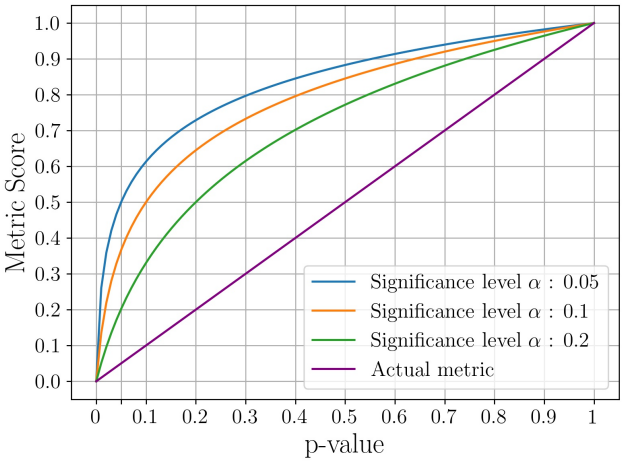

Figure 10: Statistical metric definition based on the p-value p and significance level α

3.3.3 New metrics proposal

3.3.3.1 Multivariate study : The multivariate two-sample problem

Non-parametric tests are a family of statistical tests that make no assumptions about underlying distributions. This test family is commonly used to work with multivariate distributions datasets. As one has seen, the only statistical tests implemented are based on marginal distributions. Therefore, we looked to propose multivariate statistical tests that consider entire datasets (set of columns). We used non-parametric statistical tests for this. However, because these tests are distribution-free, it is not possible to directly derive a $p-value$ from the statistics. As explained by Efron & Tibshirani in [11], **the permutation method** is one statistical trick for building a $p-value$ for any non-parametric test. The procedure is described as follows for the two sample problems: Let $\mathscr{D}_1$, $\mathscr{D}_{1,1}\cdots\mathscr{D}_{1,k}$ datasets drawn from the same distribution F_1, with $k \in \mathbb{N}$ large. Let $\mathscr{D}_2$ drawn from F_2 be the dataset that we want to perform the two sample problems, i.e. test $\mathscr{H}_0 : F_1 = F_2$. Compute the non parametric statistic $\phi(\mathscr{D}_1,\mathscr{D}_2)$ as well as the statistics $\phi(\mathscr{D}_1,\mathscr{D}_{1,i}) \forall i \in 1,\cdots,k$. Compute the $p-value$ p as $p = \frac{1}{k}\sum_{i=1}^{k} \mathbb{I}(\phi(\mathscr{D}_1,\mathscr{D}_2) < \phi(\mathscr{D}_1,\mathscr{D}_{1,i}))$ with $\mathbb{I}(x)$ the identity function, that is 1 if x is True and 0 otherwise. The main constraint of the permutation test is lack of access to a large number of datasets $\mathscr{D}_{1,1}\cdots\mathscr{D}_{1,k}$. To solve this, $\mathscr{D}_{1,1}\cdots\mathscr{D}_{1,k}$ is usually created by shuffling and sampling multiple times from a big dataset $\mathscr{D}_3$ drawn from F_1. The other constraint is the computational intensity of computing k time the statistics with k large. To overcome this, it is necessary to study the $p-value$ variation function of k for all datasets and find the first k that gives a robust $p-value$ (e.g. one that does not vary much if we repeat the same experiment).

Now that we have a way to compute a $p-value$ for a non parametric test, we present 4 non parametric tests built to solve the multivariate two sample test.

3.3.3.2 Friedmaan-Rafsky test Friedman & Rafsky [12] introduced procedures for the nonparametric two-sample problem that are based on the minimal spanning tree (MST). This test is the multivariate extension of the one defined by Wald & Wolfowitz in [13]. Let consider two columns of continuous variable A and B. This univariate test consists of sorting the elements from A combined with B and counting the number of switches r of either 1's or 2's, where each element of A is replaced by a 1 and each member of B is replaced by a 2. The null hypothesis is equivalent to a random order of 1's and 2's. As explained in [14], for the mutlivariate case, they replace the notion of a sorted list of elements from A union B with a minimal spanning tree (MST). The statistics are defined by Eq.4-7 with d_i the degree of node i in the MST, i.e. the number of edges connected to node i.

$$\phi_R = \frac{r-\mu}{\sigma} \sim \mathbb{N}(0,1) \tag{4}$$

$$\mu = \frac{2nm}{n+m} \tag{5}$$

$$\sigma^2 = \frac{2nm}{2(n+m)(n+m-1)} \cdot [(\frac{2nm-n-m}{n+m}) + \frac{c-n-m+2}{(n+m-2)(n+m-3)}[(n+m)(n+m-1)-4nm+2]] \tag{6}$$

$$c = \frac{1}{2}\sum_{i=1}^{n+m} d_i(d_i-1) \tag{7}$$

3.3.3.3 Energy test Zech & Aslan in [15] proposed a multivariate two-sample test based on the concept of minimum energy. They consider the real dataset $\mathscr{D}_r : \mathbf{X}_1, \cdots, \mathbf{X}_n$ of n samples as a system of positive charges of charge $\frac{1}{n}$ each, and the synthetic dataset $\mathscr{D}_s : \mathbf{Y}_1, \cdots, \mathbf{Y}_m$ as a system of negative charges of charge $-\frac{1}{m}$ each. The charges are normalized such that each sample contains a total charge of one unit. From electrostatics, we know that in the limit of where n, m tend to infinity, the total potential energy of the combined samples computed for a potential following a one-over-distance law will be minimum if both charge samples have the same distribution. The energy test generalizes these conditions. For the two-sample test, they use a logarithmic potential in $\mathbb{R}^d$ ($\mathbf{X}_i, \mathbf{Y}_j \in \mathbb{R}^d, \forall i \in \{1, \cdots, n\}, \forall j \in \{1, \cdots, m\}$). Their test statistics ϕ_E consist of three terms, which correspond to the energies of $\mathscr{D}_r$ ($\phi_{\mathscr{D}_r}$), $\mathscr{D}_r$ ($\phi_{\mathscr{D}_s}$) and the interaction energy ($\phi_{\mathscr{D}_r,\mathscr{D}_s}$) of the two samples. Its definition is given Eqs.8-11 with $R(r) = -ln(r)$.

$$\phi_E = \phi_{\mathscr{D}_r} + \phi_{\mathscr{D}_s} + \phi_{\mathscr{D}_r,\mathscr{D}_s} \tag{8}$$

$$\phi_{\mathscr{D}_r} = \frac{1}{n^2}\sum_{i<j} R(|\mathbf{x}_i - \mathbf{x}_j|) \tag{9}$$

$$\phi_{\mathscr{D}_s} = \frac{1}{m^2}\sum_{i<j} R(|\mathbf{y}_i - \mathbf{y}_j|) \tag{10}$$

$$\phi_{\mathscr{D}_r,\mathscr{D}_s} = -\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} R(|\mathbf{x}_i - \mathbf{y}_j|) \tag{11}$$

3.3.3.4 KNN test The K-Nearest Neighbors Test was first defined by Schilling in [16]. Taking sample $\mathscr{D}_r, \mathscr{D}_s$ the idea is to compute the distance between all real and synthetic samples. Then,

for every samples we look at its k nearest neighbors if they are real or synthetic and we expect this to be a mix of the two. If not, this means that real samples are close together and far from synthetic samples, showing a difference in the distribution. Formally, one defines $Z : \mathbf{Z}_1, \cdots, \mathbf{Z}_{n+m}$ with $\mathbf{Z}_i = \mathbf{X}_i, \forall i \in \{1, \cdots, n\}$ and $\mathbf{Z}_j = \mathbf{Y}_{j-n}, \forall j\{n+1, \cdots, n+m\}$. Let $\|\cdot\|$ be a norm and define the $k-th$ nearest neighbor to Z_i as that point Z_j satisfies $\|Z_i - Z_{j'}\| < \|Z_i - Z_j\|$ for exactly $k-1$ values of j' ($1 \leqslant j' \leqslant n+m$, $j' \neq i, j$). The first statistic proposed in [16], ϕ_k is given in Eq.12 and is simply the proportion of all k nearest neighbor comparisons in which a point and its neighbor are members of the same sample. $I_i(r) = 1$ if the $r-th$ nearest neighbor of Z_i belongs to the same sample as Z_i, $I_i(r) = 0$ otherwise. Once again, a low statistical value suggests that the two samples comes from the same distribution. To compute a $p-value$ from this statistic, the permutation method is used.

$$\phi_k = \frac{1}{(n+m)k} \sum_{i=1}^{n+m} \sum_{r=1}^{k} I_i(r) \tag{12}$$

3.3.3.5 Maximum Mean Discrepancy test (MMD) From the work of Gretton *et al.* in [17] and Borgwardt *et al.* in [18], MMD is a distance measure between distributions. Its definition is $\text{MMD}(F_1, F_2) = ||\mathbb{E}_{\mathscr{D}_r \sim F_1}(\psi(\mathscr{D}_r)) - \mathbb{E}_{\mathscr{D}_s \sim F_2}(\psi(\mathscr{D}_s))||_{\mathscr{H}}$ with $\mathscr{D}_r, \mathscr{D}_s$, random variables following F_1,F_2 distribution respectively, $||\cdot||_{\mathscr{H}}$ a norm in the Hilbert space (probability measure), ψ a function. If one defines a kernel function as a scalar product over the Hilbert space, i.e. $k(x,y) = <\psi(x), \psi(y)>_{\mathscr{H}}$, then after some computation one has $\text{MMD}^2(F_1, F_2) = \mathbb{E}_{\mathscr{D}_r, \mathscr{D}'_r \sim F_1} k(\mathscr{D}_r, \mathscr{D}'_r) + \mathbb{E}_{\mathscr{D}_s, \mathscr{D}'_s \sim F_2} k(\mathscr{D}_s, \mathscr{D}'_s) - 2\mathbb{E}_{\mathscr{D}_r \sim F_1, \mathscr{D}_s \sim F_2} k(\mathscr{D}_r, \mathscr{D}_s)$. This equation follows the kernel trick in the sense that one no longer needs to compute ψ, which can be difficult. One example of kernel function is the Gaussian given by $k(x, x') = e^{\frac{-||x-x'||^2}{2\sigma^2}}$. Here, the statistic is the square root of the empirical version of the previous equation, defined by by Eq.13. Here again, the lower the statistic, the harder is it to reject the hypothesis that both samples come from the same distribution. The permutation method is used to compute a $p-value$ for this test.

$$\phi^2_{MMD} = \frac{1}{n(n-1)} \sum_{i \neq j}^{m} k(X_i, X_j) + \frac{1}{m(m-1)} \sum_{i \neq j}^{m} k(Y_i, Y_j) - \frac{2}{nm} \sum_{i,j=1}^{m,n} k(X_i, Y_j) \tag{13}$$

3.3.4 New metrics based on ML

SDMetrics proposes machine learning metrics, including detection and efficacy metrics. For the detection metric, a classifier is trained to distinguish real and synthetic data, with the metric equal to one minus the accuracy reach of the model. The efficacy metrics describe the accuracy reached by a model trained with synthetic data and tested on real data. Here, we explore totally different fields of machine learning in order to propose new metrics. The first is based on Shapley value [19], a solution concept first used in cooperative game theory that has begun to be used in machine learning for data and features importance. The second is based on dimension reduction with principal component analysis (PCA/FAMD)

3.3.4.1 Shapley value Lloyd Shapley won the Nobel price in economics in 2012 for the introduction of the Shapley value [19]. Shapley value is a concept that tries to answer the following question : In a cooperative game where each player plays together to reach an overall gain, how important is each player to the overall cooperation, and what payoff can he or she reasonably expect? Following the work of Ghorbani & Zou[20] and also [21] [22], the players are the data rows in the training. Therefore, a Shapley value tries to estimate which rows in the training help or harm the model. Let D be the training dataset, and $V(A)$ the performance reached when the ML model is trained over dataset A. We use the notation $\phi_s(i)$ and $\phi_S(i,x)$ to define the Shapley values of row i or of row i with performance x, respectively. The Shapley values ϕ_S are defined following these rules :

- If a data row does not change the performance when added to any subset of the training data sources, then it should be given zero value. More precisely, suppose for all $S \subseteq D - \{i\}$, $V(S) = V(S \cup i)$, then $\phi_S(i)$= 0.
- If for data i and j and any subset $S \subseteq D - \{i, j\}$, we have $V(S \cup \{i\}) = V(S \cup \{j\})$ then $\phi_S(i) = \phi_S(j)$. In other words, if i and j contribute equally when added to any subset of our training data, then i and j should be given the same value by symmetry.
- When the overall performance score is the sum of separate performance scores, the overall value of a datum should be the sum of its value for each score: $\phi_S(i, V + W) = \phi_S(i, V) + \phi_S(i, W)$ for performance scores V, W. For instance, if datum i contributes values $\phi_S(i, V_1)$ and $\phi_S(i, V_2)$ to the predictions of test points 1 and 2, respectively, then we expect the value of i in predicting both test points—i.e. when $V = V1 + V2$—to be $\phi_S(i, V_1) + \phi_S(i, V_2)$.

It has been shown that any function satisfying those rules must have the shape $\phi_S(i) = c \sum_{S \subseteq D-\{i\}} \frac{V(D)-V(S)}{\begin{pmatrix} n-1 \\ |S| \end{pmatrix}}$ with c a constant and n the number of rows (datum) in D. Because this definition computationally strenuous, Ghorbani & Zou [20] proposed a Monte Carlo-based algorithm to estimate it. We use this algorithm to compute Shapley values in this thesis.

Based on the Shapley values, we propose 2 metrics to assess synthetic data quality. Both are built around the question : Does a synthetic data row help or harm the training? The first one, M_{SHAP}, is based on the mean values of synthetic and real Shapley values. To compute it, one first maps the mean of synthetic and real Shapley values $\bar{\phi}_{s,synt}, \bar{\phi}_{s,real}$ to [0,1] following a common rule, defining $\hat{\phi}_{s,synt}$, $\hat{\phi}_{s,real}$ and finally using the function $S_\alpha(p)$. We use $S_\alpha(p)$ for its property of $S_\alpha(\alpha) = 0.5$ which is also convenient in this case. The complete definition of M_{SHAP} is given in Eq.14. The second one, O_{SHAP}, is based on the magnitude order of the Shapley values. Because the higher the Shapley value is, the more the row contributes positively during training, this metric gives a positive reward if synthetic data has high Shapley values and negative rewards otherwise. To do so, the Shapley values are ordered according to their value: let $\{\bar{\bar{\phi}}\}_{i=1}^{2n}$ the set of ordered Shapley

values with $2n$ samples, n real and synthetic respectively. O_{SHAP} is defined by Eq.15 with $\mathbb{I}(\bar{\bar{\phi}}_i \in s)$, that is one if sample i is synthetic and zero otherwise.

$$M_{SHAP} = S_{\hat{\phi}_{s,real}}(\hat{\phi}_{s,synt}) = \frac{\ln(m\hat{\phi}_{s,synt}+1)}{\ln(m+1)} \quad m = \frac{1-2\hat{\phi}_{s,real}}{\hat{\phi}^2_{s,real}} \tag{14}$$

$$O_{SHAP} = \frac{1}{2} \cdot (1 + \frac{2}{n(n+1)} \sum_{i=0}^{2n} (i-n)\mathbb{I}(\bar{\bar{\phi}}_i \in s)) \tag{15}$$

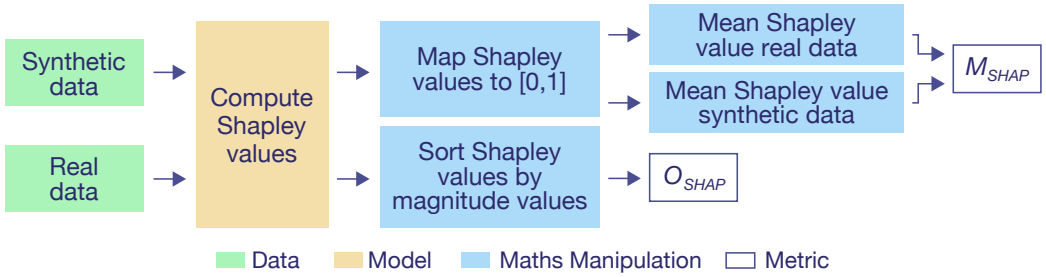

Figure 11: Scheme computation Shapley based metrics

3.3.4.2 PCA based metric PCA of a collection of points in a real coordinate space are a sequence of p unit vectors, where the $i-th$ vector is the direction of a line that best fits the data while being orthogonal to the first $i-1$ vectors. [23] It was introduced by Pearson [24] in 1901 and is now commonly used, especially in data compression. Indeed, this method allows us to reduce the dimensionality and therefore the size of the data while keeping its main information. By construction, the first dimension of a PCA captures the main information of the data. Therefore, the question motivating this metric is: Do synthetic and real data share the same main information? More specifically, if one projects them into the first principal components, will they fill the same space? We asked this question considering only the 2 first principal components because this allows us to create graphs in order to better visualize what is going on. We then used the Jaccard similarity metric [25] to assess the similarity between synthetic and real data in the reduced space. This score is the ratio between the intersection and the union of 2 spaces. It is defined between 0 and 1 and suits our assumption: the score is one when real and synthetic data fill the same subspace and share the same main information, and 0 if the scatter plots don't overlap, suggesting an important difference between the datasets. Since we are working with a dataset composed of continuous and categorical columns, we used FAMD as PCA techniques. This last is better suited for our case, and a description is given in Annexe. Then, to compute the metric, we proceed as follows: First, we fit the principal components over the real data. Next, we project the real and synthetic data according to those components, giving a scatter plot for each. Finally, we compute the area of the intersection A_I of the scatter plot and the area of their union A_U to define the metric $M_{FAMD} = \frac{A_I}{A_U}$

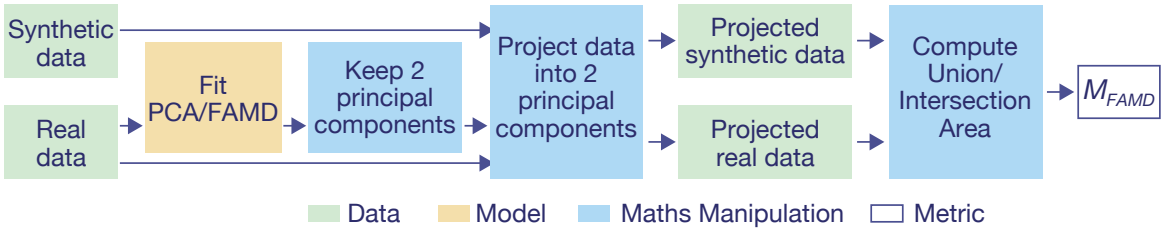

Figure 12: Scheme M_{FMAD} computation

3.4 Results *SDMetrics* review

3.4.1 Result assessment over real data

The theoretical/Implementation review has lead to corrections for the CS and KSTest. A description of the corrections that we made as well as some results deriving from these corrections are given in Section.4.4.2. For the other metrics, the theoretical definition as well as the implementation seem correct. **We pursue this review by giving only real data to all *SDMetrics***. That is to say, rather than doing `metric(real_data,synthetic_data)`, we do `metric(real_data_1,real_data_2)`. We expect this score to be high. We did this for every selected dataset. Then we took the mean score and the standard deviation among the datasets.

We fixed a threshold to 0.5 to discard metrics. If the mean among datasets is below 0.5, this means the metric is not able to give a good score even if it received only real data. Thus we cannot expect it to give a good score for "good" synthetic data. The results are presented in Fig.13 ofr all *SDMetrics*, using abbreviations. From this graph, most of the metric give a good score which prove their relevance. For the one that do not give a good score, we have the regression metrics (LR and MLPR) that are designed for regression task and should not be computed when the target is categorical. Thus it was expected to reject them in this case. Then CKLD as well as BMLPC have also a mean below 0.5 and so will not be used for the last part of the project.

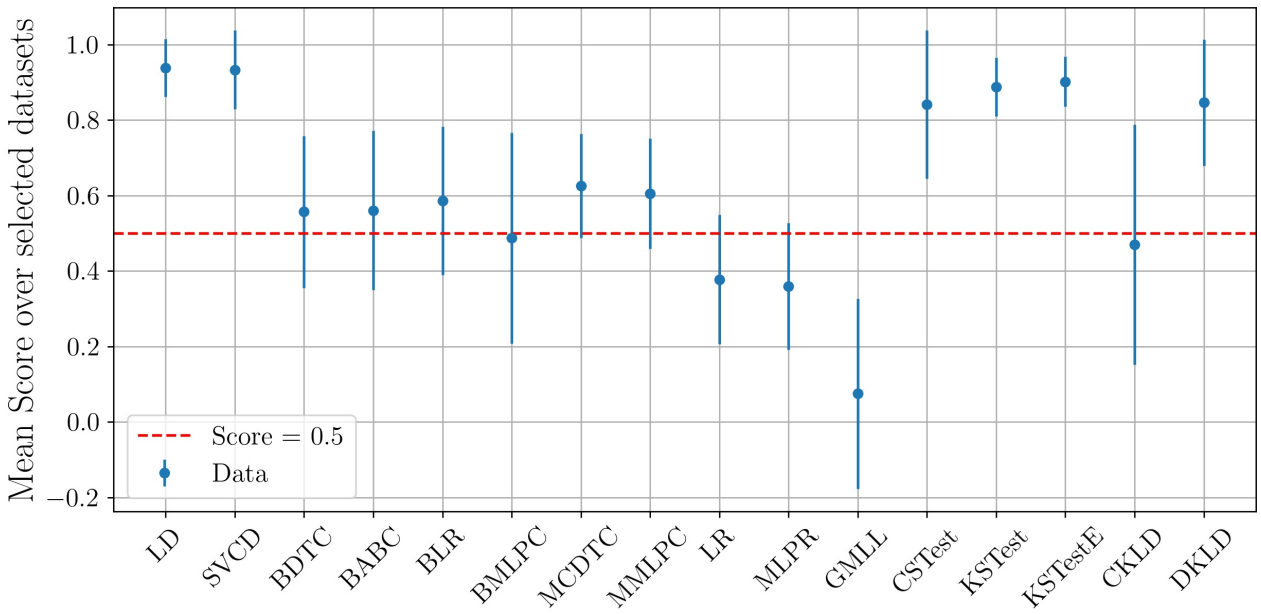

Figure 13: Mean metric score and standard deviation when only real data is given to the metrics.

3.4.2 Robustness check

The robustness check is performed to ensure that we can trust the output score of the metrics. We did this because several metrics are built from models that need to be train and depending on the training, the metrics score may change.

Therefore, for each dataset, we give thirty times the same real and synthetic data and compute the standard deviation of the metric. We then took the mean and std of this standard deviation across the datasets. Fig.14 gives the result of this experiment. From this graph, we see that the statistical metrics have no variation as expected and the metrics build on models have variations. The mean standard deviation for SVCD is 0.05 corresponding to 5% for instance. BMLPC is the metric with the highest mean variation and presents significant variation across the datasets. However putting a threshold at 0.1, all the metrics have a mean under it and can then be considered robust. No metric is discarded from this experiment.

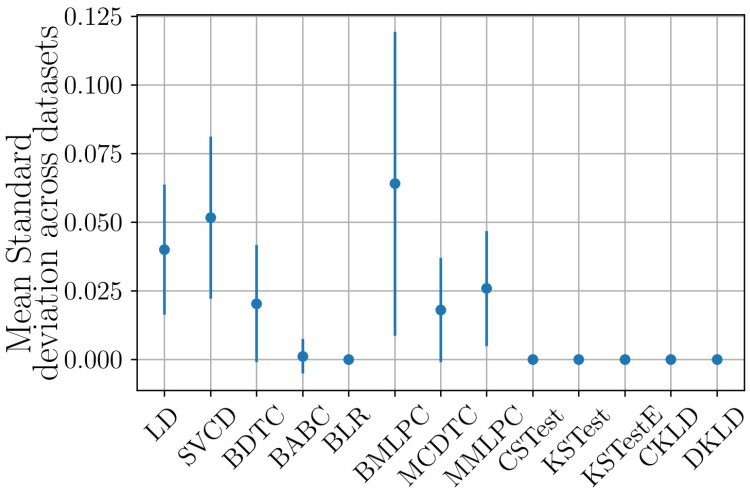

Figure 14: Mean standard deviation of the *SDMetrics*.

3.4.3 Correlation analysis

We end this metric review with a correlation analysis. The goal of this is to see if some *SDMetrics* have common output when receiving same datasets. This can be a way to reduce the number of

computed metrics.

For this experiment, we compute the metrics for all the selected datasets giving a vector of size 152 for every metric and we then compute the correlation between the vectors. The result is given in Fig.15. As we can see, all the TSTR metrics are highly correlated. This confirmed the limitation we made about redundant information of the metrics. Therefore, if we need to reduce the number of metrics, we should keep just one TSTR.

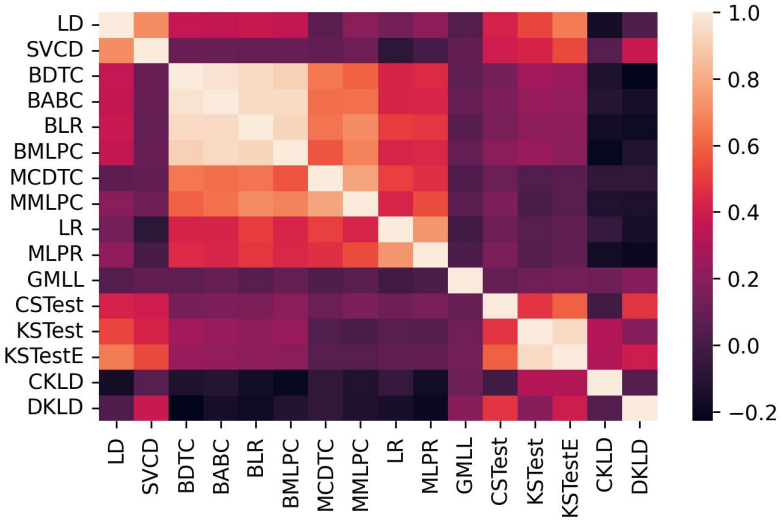

Figure 15: Heatmap correlation of*SDMetrics*.

3.5 Results new metrics proposal

3.5.1 Statistical tests

As discussed, the multivariate two sample tests will allow to consider multivariate distribution while only the marginals were considered before. Nevertheless, because they are non-parametric test, we have to built empirically a $p-value$ for them. In the Literature the **Permutation Method** (Section 3.3.3.1) is commonly used for this purpose. With this method, one parameter needs to be optimized, the number of permutation, or number of try k. The more permutation, the more robust the $p-value$ is but also the more time it takes to compute it. Here we tried to find this trade off between computational time and robustness. To do this, we increase k and for each k we compute the $p-value$ multiples times (30).

We then compute the standard deviation of the $p-value$ for this k and for the dataset used. Finally, we do this for all datasets and aggregate the result to have the mean standard deviation over the dataset for each k as well as the standard deviation of the mean. The result are presented in Fig.16. As expected the standard deviation decreased by increasing k, so the $p-value$ is more and more robust. From Fig.16, we chose $k=400$ as number of permutation because the mean variation is low enough for our purpose. Indeed we are note interested in rejecting the null hypothesis with a significance level α of 1% but rather 10% or 20% so a variation around 1% is acceptable.

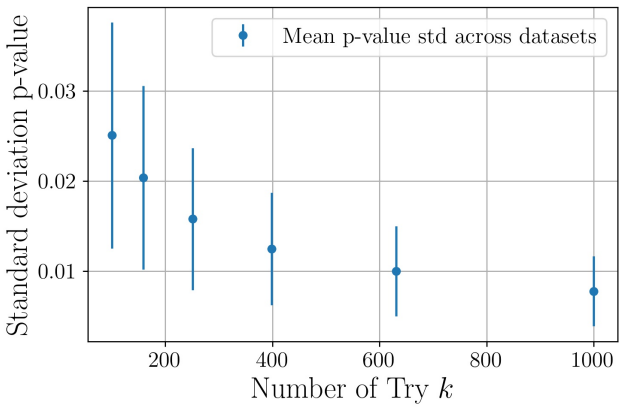

Figure 16: Mean standard deviation of the $p-value$ computed by the permutation method function of the number of permutation k.

3.5.2 Shapley value

For the metrics M_{SHAP}, O_{SHAP}, we run a first experiment to check their robustness. Since the shapley values are computed from a Monte Carlo algorithm, we wanted to ensure that the metrics score don't change if we compute multiple time the sahpley values. Especially we investigated the effect of shuffling the samples over shapley values. To do this we compared the shapley value of each sample before and after shuffling the dataset. We did this for one dataset and the results are presented in Fig.17, 18. According to Fig.17, the shapley values are well aligned along the identity line, meaning that shuffling the samples in the dataset doesn't change a lot their value. However, those little variations are enough to break the magnitude order as visible in Fig.18. From this graph we concluded that we can not use O_{SHAP} because it will not be a robust metric.

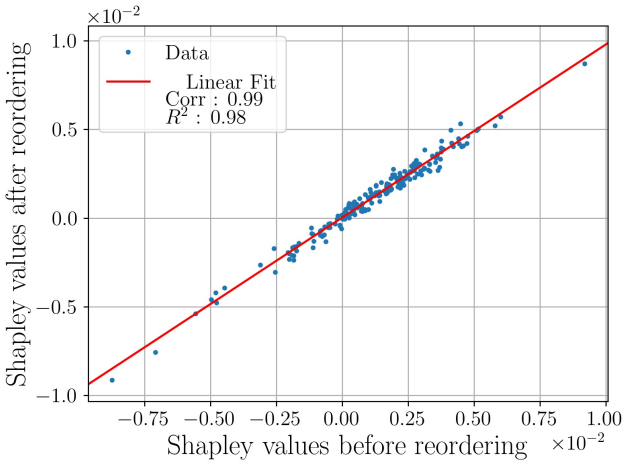

Figure 17: Shapley value variation

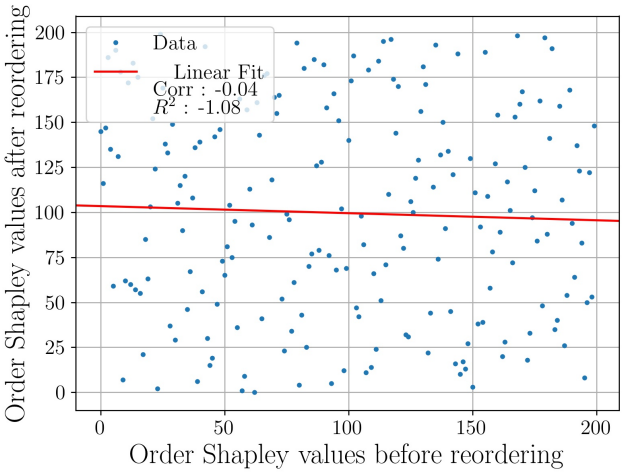

Figure 18: Shapley value order variation

Nevertheless, because the mean should not be affected by the variations shown in Fig.17, we computed it before and after shuffling the dataset and we did this for every datasets. The result presented in Fig.19 shows that the means of the synthetic or real shapley values don't vary by shuffling the sampling. Therefore as confirmed by Fig.20 M_{SHAP} is robust to the shuffling and then can be used to assess synthetic data quality.

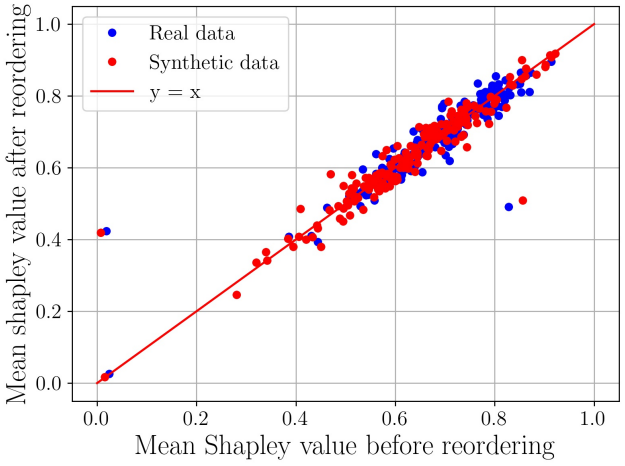

Figure 19: Mean shapley value variation

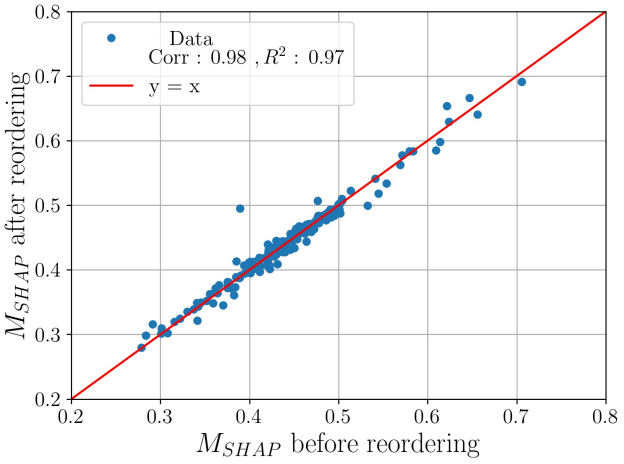

Figure 20: M_{SHAP} variation

3.5.3 PCA/FAMD based metric

For the PCA/FAMD based metric, the main challenge was to defined the area for the scatter plots. To do this, we used the `CovexHull()` method of `Scipy` to find the edge of each area. Then one difficulty was for the intersection area, since an edge is when the real and synthetic data area cross each other which is not necessarily a point of the scatter plot. To overcome this, between each edge of both area, the right equation was derived and then all the intersection points between the lines to finally isolated those that are in both area, being the intersection of the two area. The area is computed using `Polygon.area()` from `shapely.geometry` library. One example showing the different areas is given in Fig.21.

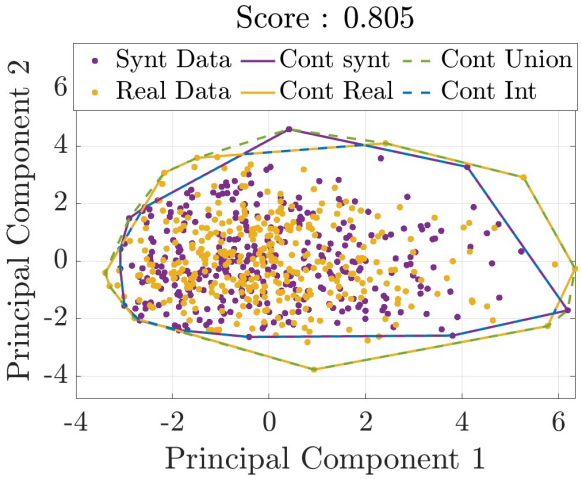

Figure 21: Example Datasets projection into principal component and metric computation

4 Model Improvement prediction based on metric score

4.1 Lack of correlation

As said before, one main use of synthetic is for model performance improvement. The data assessment should reflect this in the sense that a good synthetic data evaluation should lead to a model improvement by adding this data in the training. This is the more meaningful point of view of data assessment in this case. Of course, other limitations from the model can prevent any improvement even with a really good dataset.

This part investigates the correlation between model improvement and the final score given by *SDMetrics*. To do this, we took again many datasets (datasets informations are given in Tab.2)and generate synthetic data for each of them. We compute model performances (Accuracy, AUC, Recall, Precision and F1) before and after adding synthetic data in the training. We can then define the Improvement as : $x - a$ as the difference after and before adding synthetic data (x and a respectively). Next, for the *SDMetrics*, we use the method `evaluate(synthetic_data,real_data)` from *SDMetrics* that outputs an evaluation value of the synthetic data given the real data. This evaluation is got by aggregating the score of all the metrics described previously. In our case, the privacy metrics are not taken into account. Fig.22 is a diagram that shows how the final score of *SDMetrics* is computed. Then, doing this for all datasets, correlations between data evaluation and improvement are then computed. The result is given in Tab.4 while Fig.23 represents the improvement function of the data evaluation. Each point corresponds to one dataset. This tab shows that there is no huge correlation between the data evaluation and the model improvement. This is confirmed when looking the the scatter plot of Fig.23. Mostly, the correlation is negative in the case of the Accuracy, suggesting that a good synthetic data evaluation should lead in a decrease of the Accuracy of the model. The contrary is much more meaningful.

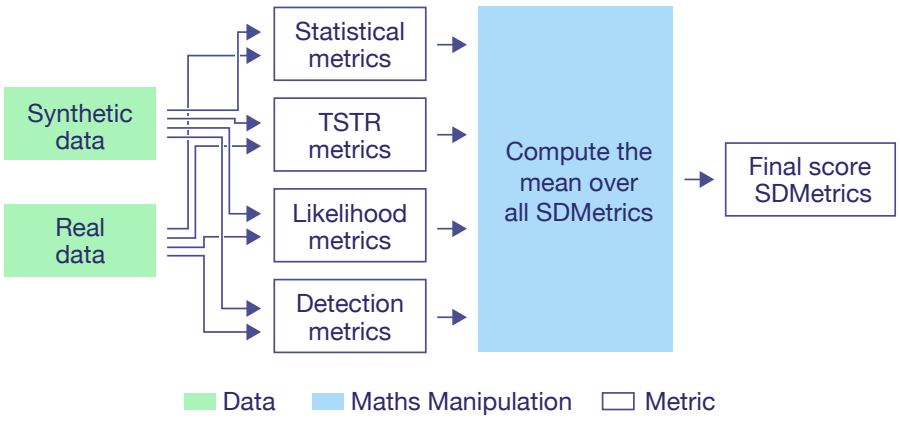

Figure 22: Computation final score *SDMetrics*

	Accuracy	AUC	Recall	Prec.	F1
Correlation Model Improvement/ *SDMetrics* score	-0.38	0.27	0.18	0.11	0.27

Table 4: Correlation between Model Improvement and score of *SDMetrics* for different model quality metrics

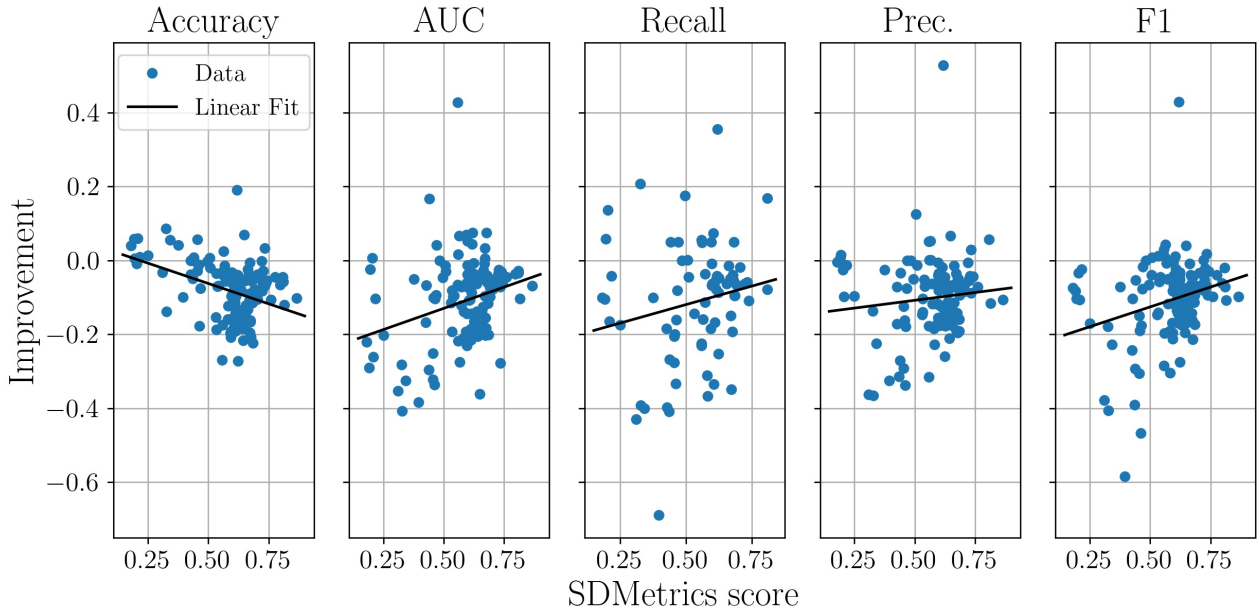

Figure 23: Correlation *SDMetrics* final score and model improvement

Finally, to better understand this result, we compute the correlation between the improvement and each metric of *SDMetrics*. As visible on Fig.24, the more dense region is near the 0 correlation, meaning that most of the *SDMetrics* have a little correlation with the model improvement. This motivates this project's main idea: to find new metrics more meaningful to assess synthetic data quality. The orange violin are got by defining a new improvement function (section 4.2) from the graph. This new definition has moved most of the correlation from a negative to a positive sign which is a good point.

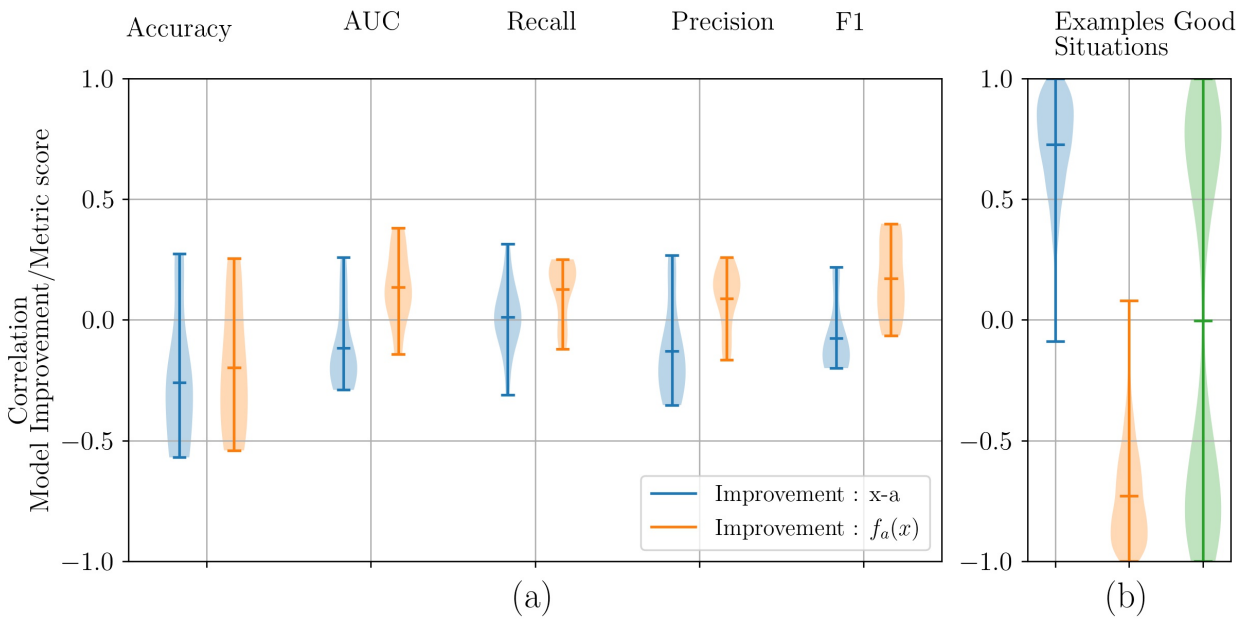

Figure 24: Density distribution of the correlation between *SDMetrics* and improvement

4.2 New Improvement method

As explained before, the model improvement is defined by Improvement : $x - a$ with a and x as the scores before and after the addition of synthetic data to the training, respectively. This definition is the most common one and is well suited for the scores we are interested in (Accuracy, Recall, F1, Precision, AUC). However, it suffers from two main limitations. First, its extreme values (i.e $[-a, 1-a]$) are not independent of a. In addition, moving the model accuracy from 0.5 to 0.53 gives the same improvement of 0.03 that moving from 0.9 to 0.93 does, while in this case an increase of 0.03 is outstanding. A new improvement function $f_a(x)$ given in Eq.16 and visible in Fig.25 was defined to overcome those limitations. The criteria that led to this new definition are as follows:

- $f_a(a) = 0$
- $f_a(x) > 0$ if $x > a$ and $f_a(x) <= 0$ if $x <= a$
- $f_a(x) \in [-1, 1] \forall x \in [0, 1]$ and continuous
- The higher a is, the better if $x > a$

$$f_a(x) = \begin{cases} \dfrac{x-a}{1-a} & x >= a \\ \dfrac{e^{\frac{x-a}{1-a}} - 1}{1 - e^{\frac{-a}{1-a}}} & x < a \end{cases} \tag{16}$$

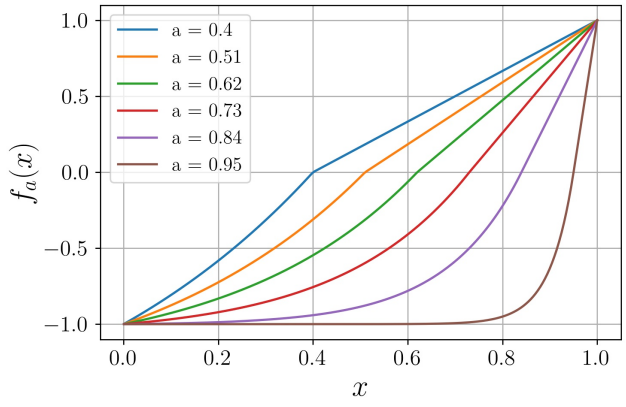

Figure 25: New Improvement function

4.3 Optimization problem

We just saw that the final result score that *SDMetrics* output is not very informative in term of model improvement. However, since synthetic data for data augmentation is used for this purpose, it could be really worth it to have a score linked to model improvement. Therefore, the idea is to try to predict the improvement of the model performance based on the metric scores. For now, as we have seen, the output $O_E(\mathscr{D}_R, \mathscr{D}_S)$ of `evaluate()` is the mean over all the computed metrics; that is $O_E(\mathscr{D}_R, \mathscr{D}_S) = \frac{1}{k}\sum_{i=1}^{k} s_i(\mathscr{D}_R, \mathscr{D}_S)$ with $\mathscr{D}_R, \mathscr{D}_S$ as the real and synthetic datasets respectively, and s_i as the metric i over the k computed. The idea is to modify this to have $O_E(\mathscr{D}_R, \mathscr{D}_S)_j = \mathscr{A}_j(\{s_i(\mathscr{D}_R, \mathscr{D}_S)\}_{i=1}^{k})$ with $\mathscr{A}_j$ as a model trained to predict the improvement of the performance j (Acc, AUC,...) ($j \in \{1, \cdots, 5\}$), receiving the metric score as input. Therefore, we now have five different outputs. Fig.26 gives a scheme of how the model $\mathscr{A}_j$ are trained from the datasets, with each color corresponding to a particular task and data type.

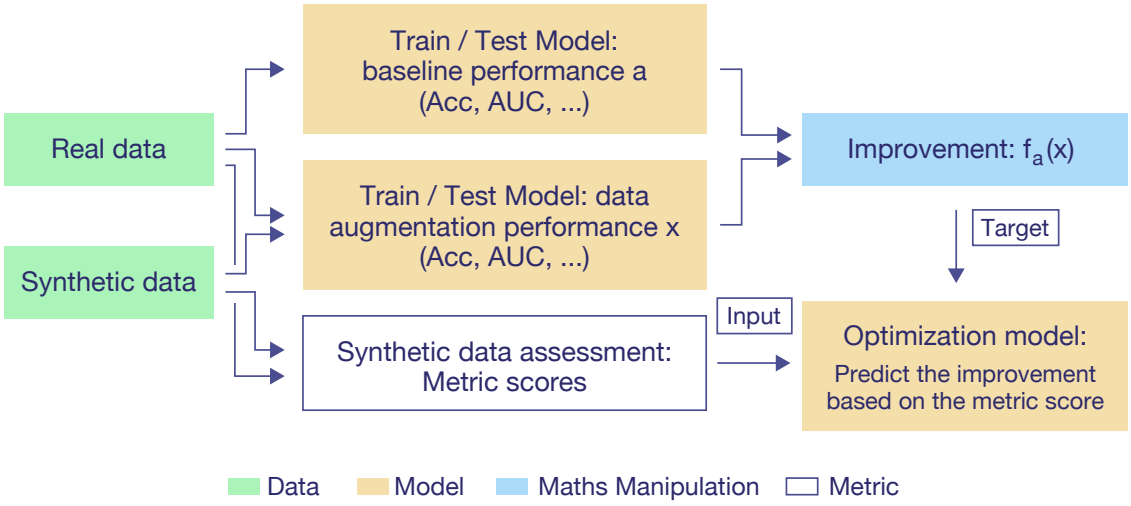

Figure 26: Optimization map

The input of every model is a dataset of shape (X, Y), with X as the number of datasets and Y as the number of metrics. The target has shape $(X, 1)$, corresponding to the improvement after data augmentation for each dataset. To do this, we used all *SDMetrics* minus those that were discarded. We added the new ones and replaced the ones we corrected. For each model performance, we then performed a model comparison to select the best model for our problem.

4.4 First results correlation metric score and model improvement

In this section, some previous results are reviewed to observe their implication in terms of model improvement. The results concerned are the dataset selection, the correction of the CSTest and KSTest of *SDMetrics*. Moreover, we propose an assessment of the new metrics based on model improvement.

4.4.1 Data selection check based on model improvement

According to Fig.6, most of the datasets must not be considered for our project (152 considered over 466). However, for the last part of the project, the optimization one, the more dataset, the better. Therefore we wanted to check that selecting those datasets was worth it. To do this, we compared the correlation between model improvement and *SDMetrics* scores when taking all the datasets or the chosen ones. We did this for all model performance scores and *SDMetrics* scores. The results are given in Fig.27. To understand this graph, the nearest to 1 or -1 the distributions are, the better in our case. This means that there is a good correlation between the SDMetric scores and the model improvement $f_a(x)$ described in Section 4.2. From Fig.27, we see that the distributions are better by selecting the dataset. Especially for the AUC or the F1 score which are the model performance of interest in our case, we moved from a mean distribution negative and close to 0 to a positive one with value around 0.2.

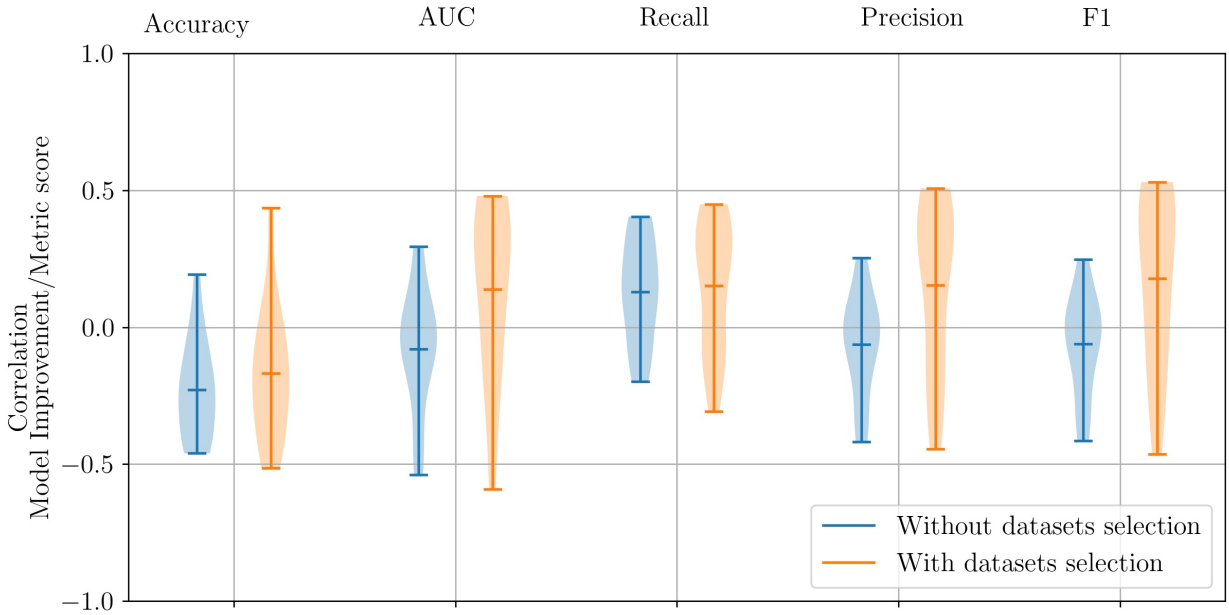

Figure 27: Violin plot presenting correlation distribution between SDMetrcis score and Model Performance score

To better understand the meaning of Fig.27, we compared the effect of the data selection for one *SDMetric*, the Binary MLPClassifier. in Fig.28 are shown the scatter plot between model improvement and metric score. One point corresponds to one dataset. For this graph, we expect to have points around the line defined by the point (0,-1), (1,1), meaning a high positive correlation, so a good metric score when the model has improved thanks to synthetic data addition. According to Fig.28, correlations seem better with the selected dataset. This is confirmed by comparing the correlation with and without selecting the dataset as done in Tab.5. From this table, selecting the datasets has increased the correlation significantly, justifying the need for this first step.

	Accuracy	AUC	Recall	Precision	F1
All dataset	**-0.44**	-0.07	0.39	-0.04	-0.02
Selected dataset	-0.38	**0.37**	**0.44**	**0.38**	**0.46**

Table 5: Comparison correlation Model Improvement/Binary MLPClassifier score before and after dataset selection

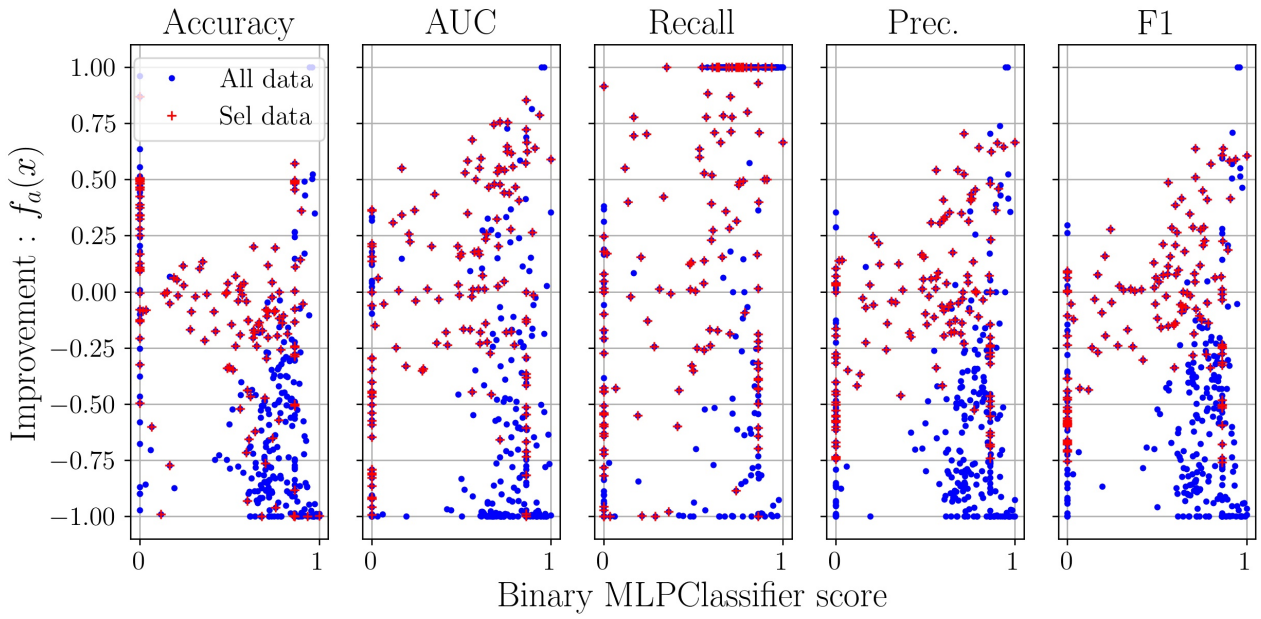

Figure 28: Scatter plot model Improvement versus Binary MLPClassifier score with all the datasets and the selected dataset.

4.4.2 *SDMetrics* corrections

4.4.2.1 CSTest The CSTest is an excellent way to check if two categorical columns come from the same distribution. However, due to a misunderstanding in `Scipy` documentation, this metric is badly computed by *SDMetrics*. In their documentation, it's written `chisquare(f_obs,f_exp)` with `f_obs,f_exp` Observed and expected frequencies in each category respectively. Nevertheless, following their tutorial, `f_obs,f_exp` appears to the observed and expected count in each category and not the frequencies. *SDMetrics* uses `scipy.stat.chisquare(f_obs,f_exp)` but gives frequencies (value between 0 and 1) while counts must be given.

We detected this error because the $p-value$ was always really high, even if the two categorical columns have totally different distributions. Therefore, we corrected this by giving counts. We then also analyzed the effect of re-mapping the $p-value$ p with $S_\alpha(p)$ choosing $\alpha = 0.1$. The results for the correlations are presented in Tab.6 as well as the scatter plot between the improvement and the metric score in Fig.29. Again in Fig.29 one point corresponds to one dataset. We can see on this graph the left shift of score made by the correction. Before due to the frequencies, the statistics was always low and so the $p-value$ really high. After the correction, a lot of score are near 0 and the $re-mapping$ allows to distribute the score less in the extremity. That was the idea behind the

$re-mapping$, to be less severe than the statistical test. From Tab.6 we see that the correction and the re-mapping have significantly increased the correlation between model improvement and the metric score. Therefore the final CSTest metric (Correction + $p-value$ mapping) is much more relevant to assess data quality for model improvement.

	Accuracy	AUC	Recall	Precision	F1
SDMetric	-0.13	-0.23	0.3	0.24	0.20
Correction	-0.18	0.46	0.39	0.42	0.41
Correction + $p-value$ mapping $S_{0.1}(p)$	**-0.19**	**0.53**	**0.46**	**0.48**	**0.48**

Table 6: Comparison correlation between Improvement function $f_a(x)$ and CSTest score before and after metrics correction and $p-value$ mapping.

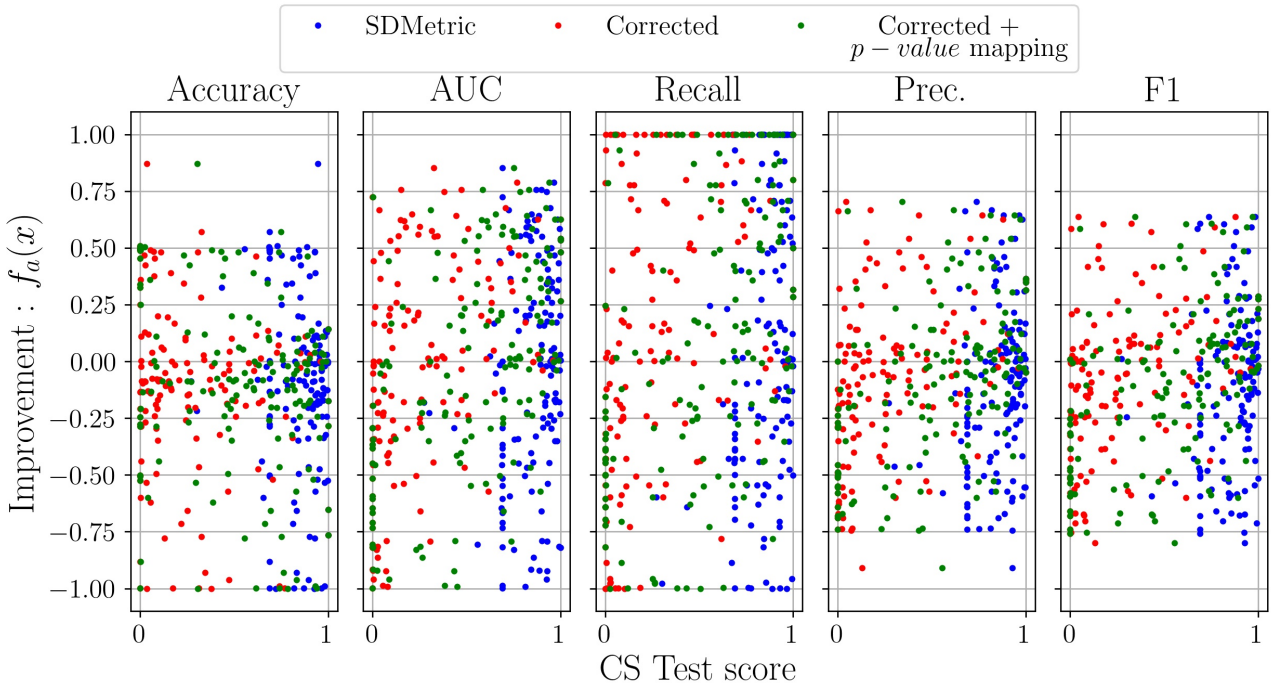

Figure 29: Comparison scatter plot Model Improvement/CS Test score for different definitions of the metric.

4.4.2.2 KSTest The KSTest is used to check if two continuous columns come from the same distribution. This time the test is well implemented but the returned score of SDMetric is a term of the test statistic rather than the $p-value$. As described in Section 3.1.1.2, the statistic is $\sqrt{m}D_m$ with m the number of samples. D_m is the maximum difference between the two ECDF and *SDMetrics* returns $1-D_m$ However by doing this, the sample size dependency of the test is totally lost. It's really different for the test and the $p-value$ to have D_m equals to 0.1 for $m=10$ or $m=1000$. Therefore the modification we made was to use the $p-value$ p of the test and to re-map it with $S_{0.1}(p)$. The results are presented in Tab.7 for the correlation and Fig.30. Here again, the correction has made increased a lot the correlation model improvement/ metric score which is good for the last part of the project. According to Tab.7, This time the $p-value$ mapping makes the correlations decrease a bit, but stay much higher than before the correction.

	Accuracy	AUC	Recall	Precision	F1
SDMetric	-0.03	-0.14	0.00	0.15	0.15
Correction	**-0.39**	**0.58**	**0.38**	**0.49**	**0.48**
Correction + $p-value$ mapping	-0.33	0.50	0.34	0.45	0.42

Table 7: Comparison correlation between Improvement function $f_a(x)$ and KSTest score before and after metrics correction and $p-value$ mapping.

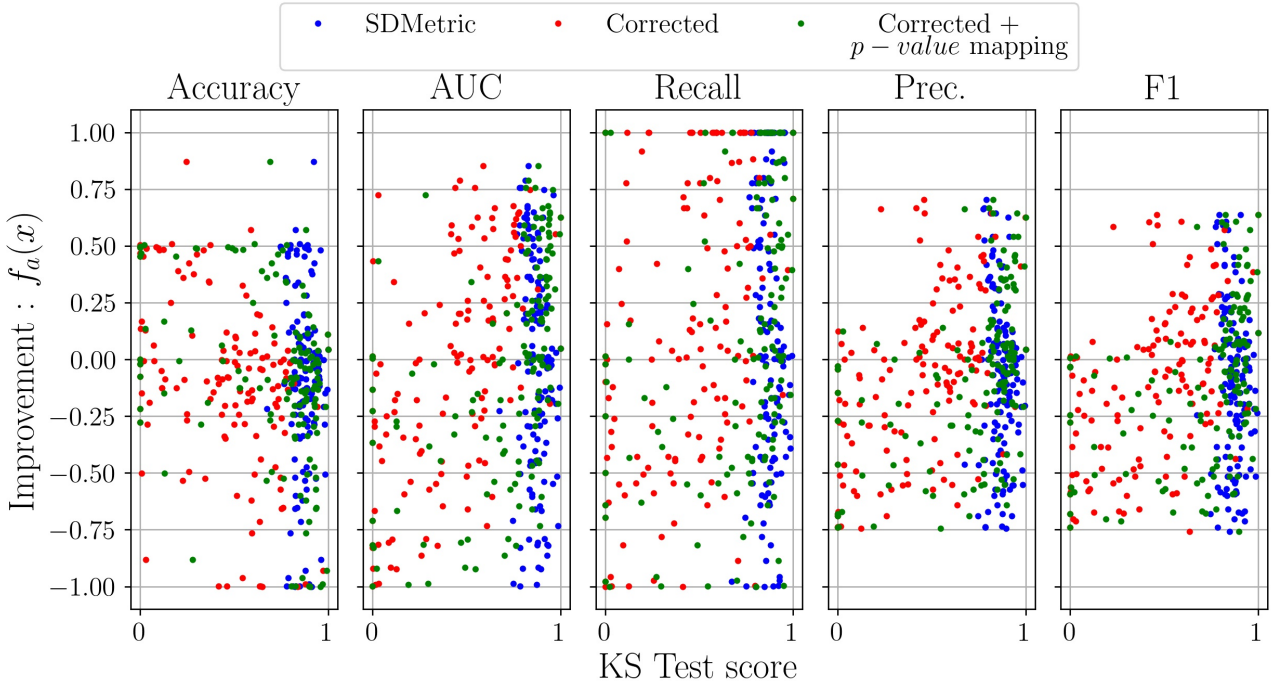

Figure 30: Comparison scatter plot Model Improvement/KS Test score for different definitions of the metric.

4.4.3 New metrics assessment

Here we compare the new metrics with the one of *SDMetrics* in terms of model improvement. The idea is then to plot again the violin distribution of the correlation between the Improvement $f_a(x)$ and the metrics score and observe where the new metrics are.

As the *SDMetrics* are diverse and more or less correlated to model improvements, we will use the following rules to assess the relevance of the new ones. If the correlation is near 0 for all performance score (Accuracy, AUC,...) the metric is not suited for our experiment. If its correlations are around the mean correlation of the *SDMetrics*, the metric is in average as good as the sdmetric which is fine. If its correlations reach high to extreme value (from 0.7 to 1 correlation in absolute value), the new metrics is particularly good for our experiment. Knowing this, we present the result in Fig.31. On this graph, one can see that the new statistical tests are always around or above the mean correlation, so they are informative M_{SHAP} is one of the most informative metric for Accuracy improvement but is anti-correlated with all other performance. Finally M_{FAMD} gives correlation a

bit lower than the other *SDMetrics* in average. However the results are good enough to use all the new metrics to build a model to predict the improvements based on metric score. This is the task of the last part of this project.

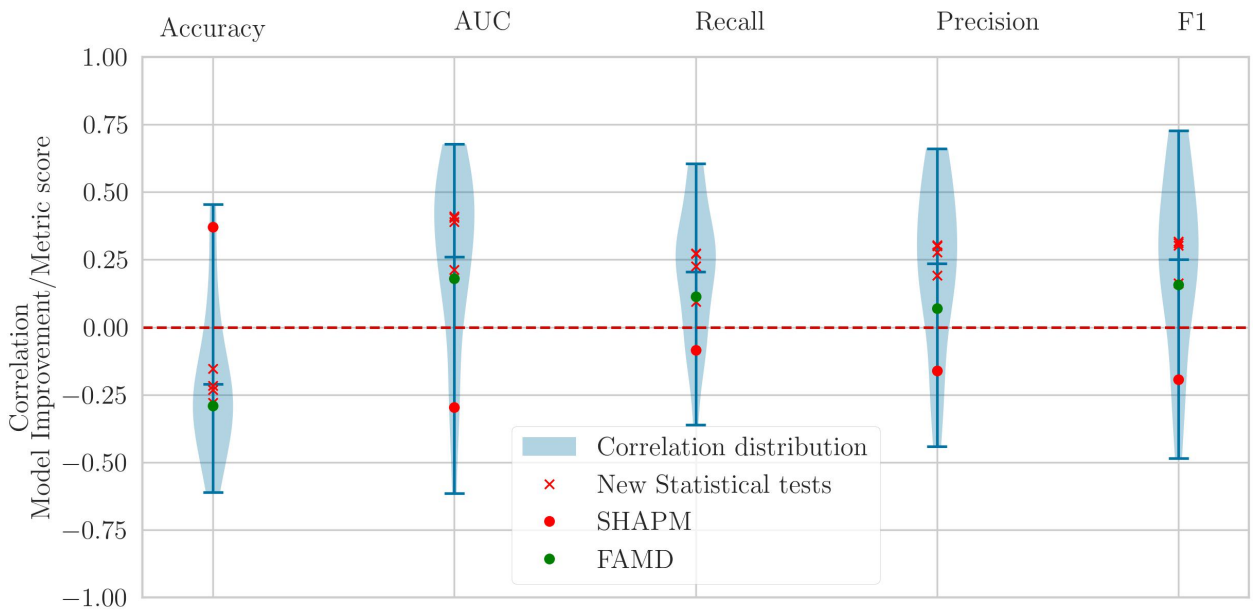

Figure 31: Correlation distributions with new metrics

4.5 Improvement prediction based on metric score

Now that we have reviewed and defined all the metrics, one can use them to solve the optimization problem. The purpose of it is to predict if a dataset (synthetic) will or not improve model performance by data augmentation based on its metric scores. When the training takes a lot of time, this reveals really useful because it will give some insights on how the model will behave by adding the dataset in the training without the need to train it with. Moreover, this is a direct approach to assess data quality based on model improvement. To perform this task, we first removed all the metrics that didn't success the first metric selection.

Then we can perform a model selection. To do this, we used Pycaret to compare many regression model architecture. Since we are doing a regression task, the performance score we're interested in are no longer the Accuracy or the Recall. The one commonly used and that we chose are given by Eqs.17-19 with y the true values, $\hat{y}$ the predicted ones and n the number of samples. $\bar{y}$, $\bar{\hat{y}}$ corresponds to the mean true and predicted values respectively. For the Mean Absolute Error (MAE) and Root Mean Squred Error (RMSE), the closer to zero, the better while for the R^2 coefficient, the closer to one the better.

$$\text{MAE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{17}$$

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{18}$$

$$R^2(y, \hat{y}) = 1 - \sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{(y_i - \bar{y})^2} \tag{19}$$

After some exploration, we decided to build one model per performance score. For each, multiple regressors are trained and tested with a cross validation. The result for the AUC are given in Tab.8 and in Appendix for the other performance score (Tabs.10-13). From those tables, Extra Trees Regressor is always the best model to fit our problem. Therefore, a quick theoretical description of it is given in Appendix. Then, the score over the cross validation are quite good, reaching a R^2 of 0.72 for the AUC for instance. This can be seen as a proof of concept that model improvement can be roughly well predicted only based on the data, without knowing the classifier or the difficulty of the task.

	Model	MAE	RMSE	R^2
et	**Extra Trees Regressor**	**0.1351**	**0.2219**	**0.7276**
lightgbm	Light Gradient Boosting Machine	0.1636	0.2331	0.6955
gbr	Gradient Boosting Regressor	0.156	0.2316	0.6941
rf	Random Forest Regressor	0.1734	0.2445	0.6621
ada	AdaBoost Regressor	0.1918	0.2499	0.6547
ridge	Ridge Regression	0.2179	0.2724	0.5892
br	Bayesian Ridge	0.2172	0.2726	0.589
lr	Linear Regression	0.2172	0.2795	0.568
dt	Decision Tree Regressor	0.171	0.2865	0.5122
knn	K Neighbors Regressor	0.2358	0.3007	0.5094
par	Passive Aggressive Regressor	0.2361	0.2955	0.5086
omp	Orthogonal Matching Pursuit	0.2517	0.3035	0.5032
lar	Least Angle Regression	0.2847	0.3593	0.0537
lasso	Lasso Regression	0.3799	0.4556	-0.1134
en	Elastic Net	0.3799	0.4556	-0.1134
llar	Lasso Least Angle Regression	0.3799	0.4556	-0.1134
dummy	Dummy Regressor	0.3799	0.4556	-0.1134

Table 8: Model comparison I AUC

To better see the meaning of the values of Tab.8, we plotted the predicted improvement versus its true improvement for one test set of the cross validation. Here again, one point corresponds to one dataset. Fig.32 presents this result for the AUC, the other performance score are given in the Appendix (Section) by Figs.35-38. As one can see on this graph, we reached a correlation of 0.8 which is satisfying and the fit we derived from the scatter plot is close to the identity line.

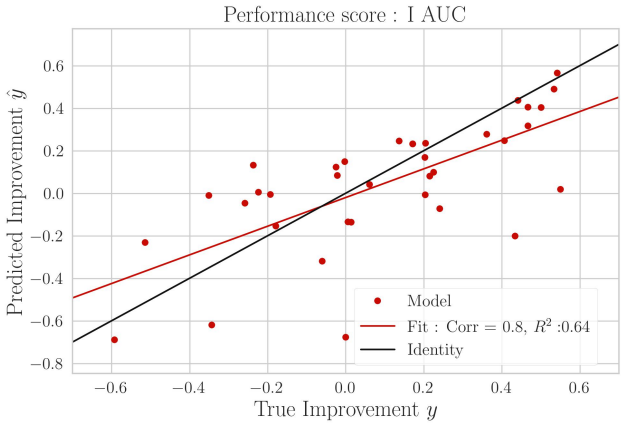

Figure 32: Comparison Model/*SDMetrics* prediction over the test set for the AUC

5 Discussion

5.1 Dataset selection

The innovative part of this project lays in how we worked with many datasets to generalize the results. Accordingly, we didn't assess the quality of a metric for one specific dataset, but over a variety of datasets. However, working with many datasets is challenging because everything we defined must work with all datasets. For this reason, we started with a pipeline to clean the datasets and generate synthetic data and meta information over the datasets (metadata to describe which category is categorical/continuous). We did not restrict the training time or architecture of the datasets (number of classes in the target column). Still, we tried to ensure that the datasets we worked needed to be augmented with synthetic data. We used the SMP (Sum of Model Performance) as a threshold to select the data. With these thresholds, we ended up using less than half of the datasets we originally had at our disposal. Fig.28 has shown that we discarded the datasets that didn't require data augmentation.

5.2 Metrics Review

Our metrics review allowed us to discover existing metrics for assessing synthetic data quality. While most of them were theoretically relevant and well implemented, we found and corrected errors in some of them. In particular, for the statistical metrics, CS and KS tests were poorly implemented, causing a loss of sample size dependency in the tests. We showed that our corrections better correlated these metrics to model improvement. The re-mapping $S_\alpha(p)$ allowed us to get more various and meaningful values for data assessment. Nevertheless, the score of those metrics is still the average over the discrete/continuous columns, therefore only based on the marginals. We tried to generalize the CSTest to a multivariate one by stacking the category, but dealing with unseen and impossible categorical configurations made these metrics uncomputable without expert knowledge. Something we didn't try that could have been interesting is to do a weighted mean over the columns. We know that for the CS Test, the more categories there are, the easier it is to reject the null hypothesis at the same sample size. Weights may correct this.

By using only real data to find the metrics, we could detect those that didn't give good results and therefore should not be used because they cannot detect real data. This allowed us to **discard some metrics** that were not designed to assess synthetic data for classification datasets. Before, those metrics had been part of the mean final score, even though they did not make sense there because they were unsuited. The Linear regression LR metric is one example of such a metric, as it is suited for regression tasks.

5.3 New Metrics proposal

The conclusions of the metric review were there was no multivariate statistical test and only two families of machine learning metrics: detection metrics and TSTR metrics (trained on synthetic data, tested on real data). Therefore, the goals were to find and implement multivariate two-sample statistical tests, and to propose other machine learning approaches to assess synthetic data.

For **the multivariate test**, we found four non-parametric tests well designed for multi-dimensional datasets. Each takes the full synthetic and real datasets as input. The challenge here was to define a $p-value$ for those tests. The permutation technique gave a robust and computationally fast solution to this challenge. We then saw that those metrics correlate with model improvement in the range of other *SDMetrics* – a bit above the mean, which is promising. However, as expected, those metrics are highly correlated because they try to check the same null hypothesis. It is good that they output similar $p-value$ because it is strange when some tests and not others reject the null hypothesis. In addition, all those metrics are based on distance between real and synthetic samples in the multi-dimensional space. While this makes total sense for continuous columns, the notion of distance between categorical distances is unclear.

The first machine learning metric we proposed is $\mathbf{M_{SHAP}}$. At first sight, the Shapley value notion perfectly fits what we are looking for. Indeed, this notion is used nowadays to assess feature and row importance in training. Because we're trying to evaluate synthetic data based on model improvement, we are particularly interested in knowing whether synthetic data rows are important or useless in the training. Therefore, in theory, the Shapley value was almost the solution for our project. However, some experimental limitations have constrained its usability. The main limit is the computational intensity required to compute the Shapley value of every row of the training set. While this is fixed by using a Monte-Carlo approach, we noticed that with this approach, we lost the order induced by the values, and thus were unable to use O_{SHAP}. The second limitation is that the Shapley values are defined according to a model, while we were looking for a model-free data assessment. In our case, we computed the Shapley values by training a logistic regression model. Last but not least, Shapley values are also defined according to one performance score. In our case, we chose Accuracy, which as is evident from Fig.31, M_{SHAP} is one of the more correlated metrics for Accuracy but is anti-correlated for other performance scores. A way to fix this could be to define multiple M_{SHAP}, one by performance score. However, this will increase the computational insensitivity, which is its main limitation. One the main challenge for using Shapley values is to fix the computational time. Indeed, to get Shapley values, one must train multiple times a model, which is time-consuming.

The FAMD metric is a totally different way to assess synthetic data. It is based on the similarity of the area spanned by the real and synthetic scatter plot in the reduced space. FAMD makes

this metric convenient for working with datasets composed of continuous and categorical columns. Moreover, this metric is fast to compute. Something that could be investigated is whether it is worth it for the training to have datasets that share the same area in the reduced space. At first sight, it would appear not to be, because this may indicate redundant information, which is bad for training. From Fig.31, we see correlations under the mean of the SDMetric,indicating it is not particularly well suited to assess synthetic data for model improvement.

5.4 Optimization problem

After checking that the metrics are meaningful for assessing synthetic data quality based on model improvement, we used them. Before this work, the metrics were aggregated by computing a mean over all of them. We showed in Section.3.2 that doing this fails to output an informative score about model improvement. Therefore, we built a regression model to predict improvement from the metric score. According to Tab.8 we reached a good R^2 coefficient. This can be seen as a proof of concept that model improvement can be assessed based only on the data, without models. To extend this, the model should be tested over more datasets and with different proportions of synthetic and real data. We only studied cases in which there is the same amount of synthetic and real data in the training because this is common, but the results could be different in other situations.

5.5 Future work

The main future work of this project is to create an end-to-end pipeline from synthetic data generation to metrics computation and data assessment, thanks to the last model. This pipeline will be user-oriented, in that the users should be able to specify which model they want to improve according to which performance score.

Research into new metrics is still a wide open field. One could first try to improve the use of the Shapley value notion to be less demanding computationally and more robust. O_{SHAP} is an interesting metric if we can classify samples in a robust way. Now that we have a model that is very good at predicting the improvement from the metrics, an interesting idea could be to add the output of the model as a loss term in the data generators. Thanks to this, a data generator may be able to generate better synthetic data, at least in the sense of model improvement. The first way to do this is by considering only one performance score (ex. AUC), adding the output of the model in consideration, and checking its improvement at each step of the training/generation. In order to impact the training, the adding term to the loss function should be weighted, and the weight optimized.

6 Conclusion

This project was a deep investigation into data assessment. Motivated by the need for data and to improve ML models' performances, we focused on synthetic data assessment based on model improvement. Our final goal was, given a real and a synthetic dataset, to evaluate whether it is worth adding the synthetic dataset to the train set to improve model performance.

To address these challenging questions, we used a large number of datasets from various fields to generalize the application domain of this work. We sought to propose an evaluation of the data that was as independent as possible from the size and type of dataset and the models used. To do this, we combined existing metrics for data assessment and developed new ones. All metrics involved presented a statistical or machine learning insight into the dataset. Analysis and review of existing metrics have oriented research on new metrics around the implementation of multivariate two-sample tests and ML metrics based on unexploited notions such as the Shapley value and PCA/FAMD.

After the implementation and relevance check into the new metrics, we looked to build models to predict improvement based on metric scores. We used all the metrics, and achieved a proof of concept that model improvement can be assessed based only on the data.

References

1. Lapedes, D. N. McGraw-Hill dictionary of scientific and technical terms. Second edition. English. Publisher: McGraw-Hill Book Co.,New York, NY. https://www.osti.gov/biblio/6451234 (2022) (Jan. 1978).
2. *The real promise of synthetic data* en. https://news.mit.edu/2020/real-promise-synthetic-data-1016 (2022).
3. Goodfellow, I. *et al. Generative Adversarial Nets* in *Advances in Neural Information Processing Systems* **27** (Curran Associates, Inc., 2014). https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html (2022).
4. *Which Face Is Real?* https://www.whichfaceisreal.com/index.php (2022).
5. Yogev, R. *The Importance of Synthetic Data for Privacy Preservation* en_US. https://www.datomize.com/resources/the-importance-of-synthetic-data-for-privacy-preservation/ (2022).
6. al, N. P. e. *Data synthesis based on generative adversarial networks | EndNote Click* en. https://click.endnote.com/viewer?doi=10.14778%2F3231751.3231757&token=WzIyNzQ0ODMsIjEwLjE opq64HtB-fzoaDRsgPTQj-22i-4 (2022).
7. Xu, L. & Veeramachaneni, K. Synthesizing Tabular Data using Generative Adversarial Networks. *arXiv:1811.11264 [cs, stat]*. arXiv: 1811.11264. http://arxiv.org/abs/1811.11264 (2022) (Nov. 2018).
8. Patki, N., Wedge, R. & Veeramachaneni, K. *The Synthetic Data Vault* en. in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (IEEE, Montreal, QC, Canada, Oct. 2016), 399–410. ISBN: 978-1-5090-5206-6. http://ieeexplore.ieee.org/document/7796926/ (2022).
9. Nečasová, T. & Svoboda, D. en. in *Computer Vision – ECCV 2018 Workshops* (eds Leal-Taixé, L. & Roth, S.) Series Title: Lecture Notes in Computer Science, 385–394 (Springer International Publishing, Cham, 2019). ISBN: 978-3-030-11023-9 978-3-030-11024-6. http://link.springer.com/10.1007/978-3-030-11024-6_28 (2022).
10. *Measuring the quality of Synthetic data for use in competitions* arXiv:1806.11345 [cs, stat]. June 2018. http://arxiv.org/abs/1806.11345 (2022).
11. Efron, B. & Tibshirani, R. *An introduction to the bootstrap* en. *Monographs on statistics and applied probability* **57.** ISBN: 978-0-412-04231-7 (Chapman & Hall, New York, 1993).
12. Friedman, J. H. & Rafsky, L. C. Multivariate Generalizations of the Wald-Wolfowitz and Smirnov Two-Sample Tests. *The Annals of Statistics* **7.** Publisher: Institute of Mathematical Statistics, 697–717. ISSN: 0090-5364. https://www.jstor.org/stable/2958919 (2022) (1979).

13. Wald, A. & Wolfowitz, J. An Exact Test for Randomness in the Non-Parametric Case Based on Serial Correlation. *The Annals of Mathematical Statistics* **14.** Publisher: Institute of Mathematical Statistics, 378–388. ISSN: 0003-4851, 2168-8990. https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-14/issue-4/An-Exact-Test-for-Randomness-in-the-Non-Parametric-Case/10.1214/aoms/1177731358.full (2022) (Dec. 1943).

14. *Friedman-Rafsky Test | Real Statistics Using Excel* https://www.real-statistics.com/multivariate-statistics/multivariate-normal-distribution/friedman-rafsky-test/ (2022).

15. Zech, G. & Aslan, B. A Multivariate Two-Sample Test Based on the Concept of Minimum Energy. en, 4 (2003).

16. Schilling, M. F. *Multivariate Two-Sample Tests Based on Nearest Neighbors | EndNote Click* en. https://click.endnote.com/viewer?doi=10.2307%2F2289012&token=WzIyNzQ0ODMsIjEwLjIz
21jeQssGvQJYPrdMQnuzWJwekkw (2022).

17. Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. & Smola, A. A Kernel Two-Sample Test. *Journal of Machine Learning Research* **13,** 723–773. ISSN: 1533-7928. http://jmlr.org/papers/v13/gretton12a.html (2022) (2012).

18. Borgwardt, K. M. *et al.* Integrating structured biological data by Kernel Maximum Mean Discrepancy. en. *Bioinformatics* **22,** e49–e57. ISSN: 1367-4803, 1460-2059. https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btl242 (2022) (July 2006).

19. Shapley, L. S. QUOTA SOLUTIONS OP n-PERSON GAMES1. *Edited by Emil Artin and Marston Morse,* 343 (1953).

20. Ghorbani, A. & Zou, J. Y. What is your data worth? Equitable Valuation of Data. en, 23.

21. Jia, R. *et al.* Towards Efficient Data Valuation Based on the Shapley Value. en. https://arxiv.org/abs/1902.10275v3 (2022) (Feb. 2019).

22. Tang, S. *et al.* Data valuation for medical imaging using Shapley value and application to a large-scale chest X-ray dataset. en. *Scientific Reports* **11.** Number: 1 Publisher: Nature Publishing Group, 8366. ISSN: 2045-2322. https://www.nature.com/articles/s41598-021-87762-2 (2022) (Apr. 2021).

23. *Principal component analysis* en. Page Version ID: 1087053745. May 2022. https://en.wikipedia.org/w/index.php?title=Principal_component_analysis&oldid=1087053745 (2022).

24. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. https://zenodo.org/record/1430636 (2022) (Nov. 1901).

25. Jaccard, P. The Distribution of the Flora in the Alpine Zone.1. en. *New Phytologist* **11.** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1469-8137.1912.tb05611.x, 37–50. ISSN: 1469-8137. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-8137.1912.tb05611.x (2022) (1912).

26. *Precision and recall* en. Page Version ID: 1089762876. May 2022. https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=1089762876 (2022).

27. Sklar, M. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris* **8,** 229–231 (1959).

28. *Multiple correspondence analysis* en. Page Version ID: 1082089444. Apr. 2022. https://en.wikipedia.org/w/index.php?title=Multiple_correspondence_analysis&oldid=1082089444 (2022).

29. Beheshti, N. *Random Forest Regression* en. Mar. 2022. https://towardsdatascience.com/random-forest-regression-5f605132d19d (2022).

30. *What is the difference between Extra Trees and Random Forest? Quantdare* en-GB. June 2020. https://quantdare.com/what-is-the-difference-between-extra-trees-and-random-forest/ (2022).

7 Appendix

7.1 Model Evaluation

Model evaluation for classification models is easier than for regression models because there are only 4 different scenarios. A sample is a true positive in a test set if it has a label and the model predicted this label. It is a false negative if the model didn't predict the label. If the sample doesn't have a label, but the model predicted one, this is a false positive, while if it does not have a label and none is predicted, this is a true negative. Those four situations are often presented as in Tab.9. The main metrics for model evaluation are based on this table.

		Actual	
		Postive	Negative
Predicted	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Table 9: Confusion matrix

7.1.1 Accuracy

Accuracy is one of the more common metrics. It computes the ratio of good prediction over all prediction as shown by Eq.20 :

$$\text{Accuracy} : \frac{TP + TF}{TP + FP + FN + TN} \tag{20}$$

Accuracy and all the following metrics are defined between 0 and 1. For Accuracy, 0 means no predictions are correct and 1 that all are correct.

7.1.2 Recall

Recall answers the following question: What proportion of actual Positives are correctly classified? Eq.21 defines this metric, which is mainly used when capturing a positive sample is what really matters – for instance, for decease detection.

$$\text{Recall} : \frac{TP}{TP + FN} \tag{21}$$

7.1.3 Precision

Precision answers the question: What proportion of predicted Positives are truly Positive? It's defined by Eq.22 and is 1 when all the predicted positives are truly positive.

$$\text{Precision} : \frac{TP}{TP + TN} \tag{22}$$

Fig.33 illustrates the difference between Recall and Precision.

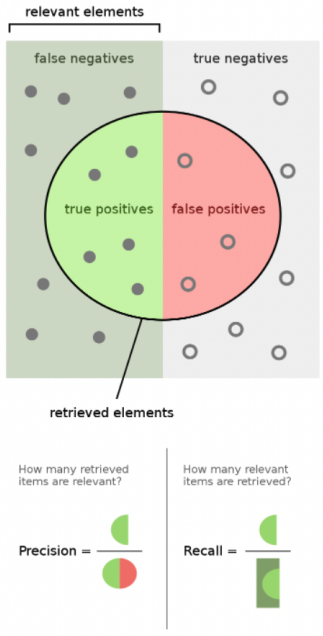

Figure 33: Illustration of Recall and Precision [26]

7.1.4 F1 Score

The F1 Score corresponds to the harmonic mean of Precision and Recall. It is defined by Eq.23 and can be seen as a balance between Recall and Precision.

$$\text{F1 Score} : 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{23}$$

7.1.5 AUC

The ROC (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. It is a plot of the Recall function of the False Positive Rate, defined as : FPR = $\frac{\text{FP}}{\text{FP} + \text{TN}}$. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. AUC stands for "Area under the ROC Curve." AUC measures the entire two-dimensional area underneath the entire ROC curve.

7.2 Copula Data Generator

Sklar in [27] proposed the first definition of copula. From his theorem, any multivariate distribution (CDF) can be written as a function of its marginal, that is to say $F(x_1, \cdots, x_n) = C(F(x_1), \cdots, F(x_n))$ with C a copula function. A copula represents the dependencies between each of the features. The key intuition underlying copula functions is the idea that marginal distributions can be modeled independently from the joint distribution. Therefore, the idea behind the copula data generator is to first fit every dataset column using univariate fitting method (Gaussian univariate) and then model the relationship between columns with another fitting method (based on correlation between columns, for instance). In this project we used the Copula Data Generator of *SDMetrics* [8].

7.3 FAMD

FAMD works as a principal component analysis (PCA) for quantitative variables and as a multiple correspondence analysis (MCA) for qualitative variables. Both achieve dimension reduction using matrix truncation. For the PCA, let us consider X_{cont} a $n \times m$ matrix containing continuous columns. The singular value decomposition of X_{cont} is $X_{cont} = U\Sigma W^T$ with U a $n \times n$ matrix, Σ a $n \times m$ diagonal matrix containing singular values of X_{cont} and W a $m \times m$ matrix. The singular value decomposition of X_{cont} is $X_{cont} = U\Sigma W^T$ with U a $n \times n$ matrix, Σ a $n \times m$ diagonal matrix containing singular values of X_{cont} and W a $m \times m$ matrix. Then, to reduce the space to a dimension $L < m$, the truncation matrix is defined by $T_L = U_L \Sigma_L$ with U_L equal to U, Σ_L a $n \times L$ matrix containing the L first column of Σ. For MCA, from [28], let us consider X_{cat} as a $n \times p$ matrix containing p categorical columns. One can first construct X a $n \times k$ matrix containing only 0 and 1. Here $k = \sum_{i=1}^{p} j_k$ with j_k as the number of category of column k. Set the sum of all entries to be b and define $Z = \frac{1}{b}X$, In an MCA, there are also two special vectors: first r, which contains the sums along the rows of Z, and then c, which contains the sums along the columns of Z. Note $D_r = \text{diag}(r)$

and $D_c = \text{diag}(c)$, the diagonal matrices containing r and c respectively as diagonal. With these notations, computing an MCA consists essentially of the singular value decomposition of the matrix $M = D_r^{-\frac{1}{2}}(Z - rc^t)D_c^{-\frac{1}{2}}$. Therefore one can write $M = P\Delta Q^T$ as singular decomposition and to the truncated matrix is $M_L = P\Delta_L$ with Δ_L the $n \times L$ with the first singular values of M. Combining the PCA and the MCA allows us to reduce the entire dataset to a given dimension L.

7.4 Extra Tree Regressor

Extra Tree Regressor is a type of Random Forest that uses multiple Decision Tree (DT). Fig.34 gives an example of DT in the case of a dataset with three features (Var_1, Var_2, Var_3), a, b, c, d being any numeric or categorical value. For a RF for regression, the output is a numerical value and the different possible outputs correspond to the circles. The three must be read from the top to the bottom. Then, a RF creates a lot of random DT and aggregates the result between them. It does this by bootstrapping, which is randomly sampling subsets of a dataset over a given number of iterations and a given number of variables. This is done to have as uncorrelated a DT as possible.

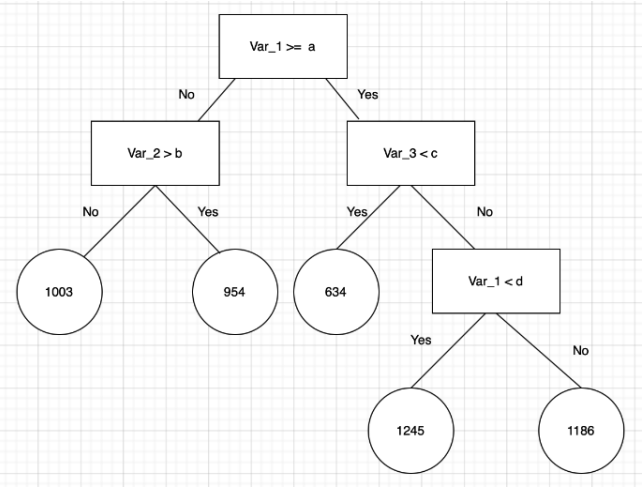

Figure 34: Example Decision Tree [29]

From [30], Extra Tree Regressor does not use bootstrapping, but the whole dataset. Another difference is the selection of cut points in order to split nodes. Random Forest chooses the optimum split while Extra Trees chooses it randomly. These differences motivate the reduction of both bias and variance. On one hand, using the whole original sample instead of a bootstrap replica will reduce bias. On the other hand, randomly choosing the split point of each node will reduce variance.

	Model	MAE	RMSE	R^2
et	**Extra Trees Regressor**	**0.121**	**0.2065**	**0.6322**
gbr	Gradient Boosting Regressor	0.1534	0.2266	0.5775
rf	Random Forest Regressor	0.1702	0.2367	0.5363
lightgbm	Light Gradient Boosting Machine	0.1804	0.2466	0.5174
ada	AdaBoost Regressor	0.2073	0.2615	0.4576
dt	Decision Tree Regressor	0.1528	0.2645	0.3385
omp	Orthogonal Matching Pursuit	0.2252	0.2996	0.2774
knn	K Neighbors Regressor	0.2369	0.3073	0.2656
ridge	Ridge Regression	0.2405	0.3128	0.238
br	Bayesian Ridge	0.243	0.3142	0.2292
lr	Linear Regression	0.2568	0.3271	0.155
lar	Least Angle Regression	0.2838	0.36	-0.03
lasso	Lasso Regression	0.2868	0.3796	-0.0913
en	Elastic Net	0.2868	0.3796	-0.0913
llar	Lasso Least Angle Regression	0.2868	0.3796	-0.0913
dummy	Dummy Regressor	0.2868	0.3796	-0.0913
par	Passive Aggressive Regressor	0.3025	0.38	-0.2052

Table 10: Model comparison I Accuracy

	Model	MAE	RMSE	R^2
et	**Extra Trees Regressor**	**0.1169**	**0.1774**	**0.6938**
rf	Random Forest Regressor	0.1363	0.1835	0.6714
gbr	Gradient Boosting Regressor	0.1327	0.1824	0.6682
lightgbm	Light Gradient Boosting Machine	0.1445	0.1916	0.6442
ada	AdaBoost Regressor	0.159	0.197	0.6223
knn	K Neighbors Regressor	0.173	0.218	0.5483
ridge	Ridge Regression	0.1822	0.2196	0.534
br	Bayesian Ridge	0.1828	0.2201	0.5318
dt	Decision Tree Regressor	0.1397	0.2124	0.4991
lr	Linear Regression	0.1947	0.2385	0.4537
omp	Orthogonal Matching Pursuit	0.2037	0.2502	0.4029
par	Passive Aggressive Regressor	0.2007	0.2514	0.3602
lasso	Lasso Regression	0.2689	0.3355	-0.0468
en	Elastic Net	0.2689	0.3355	-0.0468
llar	Lasso Least Angle Regression	0.2689	0.3355	-0.0468
dummy	Dummy Regressor	0.2689	0.3355	-0.0468
lar	Least Angle Regression	0.3845	0.4942	-1.9613

Table 11: Model Comparison I F1

	Model	MAE	RMSE	R2
gbr	Gradient Boosting Regressor	0.134	0.1869	**0.62**
et	Extra Trees Regressor	**0.1209**	**0.1834**	0.6129
rf	Random Forest Regressor	0.1423	0.1897	0.604
ada	AdaBoost Regressor	0.1631	0.2	0.5796
lightgbm	Light Gradient Boosting Machine	0.1435	0.1887	0.5744
ridge	Ridge Regression	0.1944	0.2372	0.4289
br	Bayesian Ridge	0.1948	0.238	0.4244
knn	K Neighbors Regressor	0.1915	0.2435	0.3987
dt	Decision Tree Regressor	0.1577	0.2405	0.3967
lr	Linear Regression	0.2049	0.2504	0.319
omp	Orthogonal Matching Pursuit	0.2218	0.2738	0.2206
par	Passive Aggressive Regressor	0.2341	0.2865	0.1781
lasso	Lasso Regression	0.2731	0.3429	-0.0582
en	Elastic Net	0.2731	0.3429	-0.0582
llar	Lasso Least Angle Regression	0.2731	0.3429	-0.0582
dummy	Dummy Regressor	0.2731	0.3429	-0.0582
lar	Least Angle Regression	0.7817	1.0134	-23.9121

Table 12: Model comparison I Precision

	Model	MAE	MSE	R^2
et	**Extra Trees Regressor**	**0.2498**	**0.1572**	**0.5369**
rf	Random Forest Regressor	0.31	0.1697	0.4887
lightgbm	Light Gradient Boosting Machine	0.3094	0.1739	0.48
gbr	Gradient Boosting Regressor	0.2854	0.1706	0.4751
ada	AdaBoost Regressor	0.3507	0.1884	0.436
ridge	Ridge Regression	0.3569	0.2024	0.3673
br	Bayesian Ridge	0.3598	0.2041	0.3642
lr	Linear Regression	0.3757	0.2165	0.3108
omp	Orthogonal Matching Pursuit	0.4037	0.2274	0.2917
knn	K Neighbors Regressor	0.4019	0.2485	0.2354
par	Passive Aggressive Regressor	0.4286	0.2749	0.1129
lar	Least Angle Regression	0.4506	0.2986	0.0508
dt	Decision Tree Regressor	0.3729	0.3478	-0.0604
lasso	Lasso Regression	0.5244	0.3743	-0.1127
en	Elastic Net	0.5244	0.3743	-0.1127
llar	Lasso Least Angle Regression	0.5244	0.3743	-0.1127
dummy	Dummy Regressor	0.5244	0.3743	-0.1127

Table 13: Model comparison I Recall

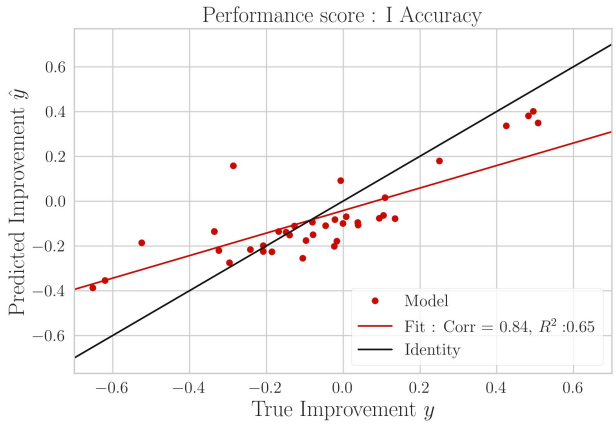

Figure 35: Comparison Model/*SDMetrics* prediction over the test set for the Accuracy

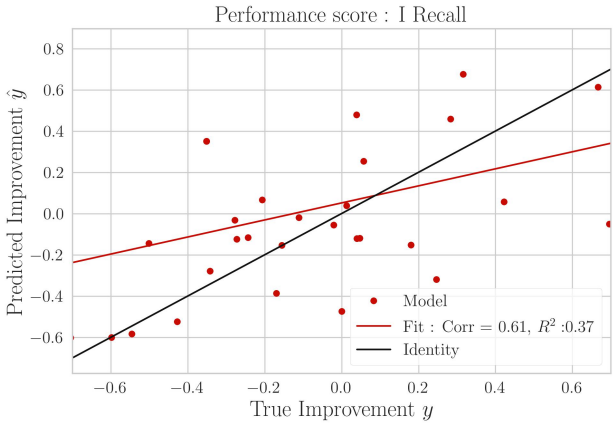

Figure 36: Comparison Model/*SDMetrics* prediction over the test set for the Recall

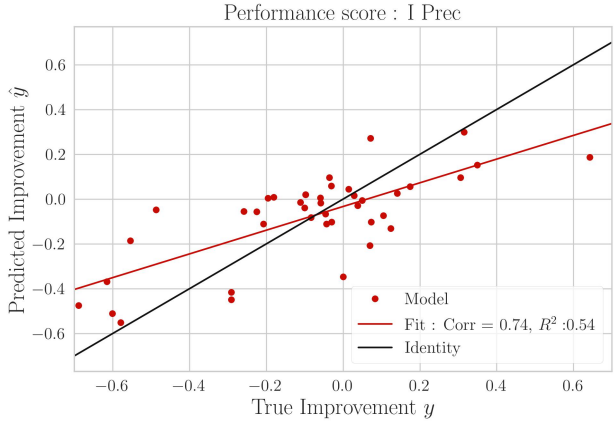

Figure 37: Comparison Model/*SDMetrics* prediction over the test set for the Precision

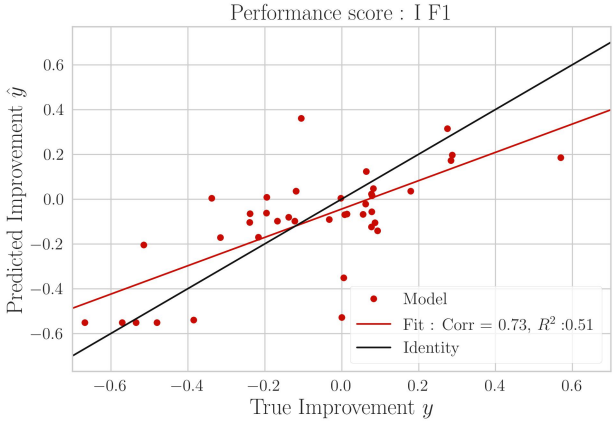

Figure 38: Comparison Model/*SDMetrics* prediction over the test set for the F1