UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

DEGREE'S FINAL DISSERTATION

Degree in Chemical Engineering

Generation and evaluation of synthetic Covid-19 patients data

Report and Appendix

Author: Alex Martin Peyrona
Director: Magda Ruíz Ordóñez
Co-Director: Luis Eduardo Mujica Delgado
Submission Date: January 17th 2023

RESUM

Aquest treball de final de grau es centra en un problema molt comú avui dia: la falta de dades. Els cientifics de dades necessiten volums grans de dades per produir eines precises i acurades.

En aquest document s'investiga sobre un conjunt de dades de pacients covid, el qual es descriu en tots els seus aspectes. L'objectiu es produir dades noves, per tant, en centrar la recerca en diferents algoritmes i models que ho permetin.

S'han triat tres algoritmes que estan basats en les copules gaussianes, reds generatives adversaries i reds bayesianes, cadascun. Es generen les dades i s'estudia com de $bones$ son.

La evaluació d'aquestes es duu a terme per dues bandes. Per una banda, mitjançant un test d'hipótesis, s'evalua si la distribució sintètica es similar (estadisticament) a la distribució original, comparant estadístics com la variança i la mediana. D'altre banda, s'evalua si es mantenen les correlacions de les dades originals a les sintètiques mitjançant un anàlisis de similaritat utilitzant matrius de coeficients de correlació. S'efectua una diferència de matrius (original i sintètica), la qual, idealment, hauria de ser molt propera a zero.

Tot plegat, s'analitzen els resultats de les dues evaluacions i es proposen idees per seguir aquesta branca de la investigació. A més, s'analitza l'impacte econòmic i ambiental del projecte

RESUMEN

Este trabajo de fin de grado se centra en un problema muy común hoy en día: la falta de datos. Los científicos de datos necesitan grandes volúmenes de datos para producir herramientas precisas.

En este documento se investiga sobre un conjunto de datos de pacientes covid, el cual se describe en todos sus aspectos. El objetivo es producir datos sintéticos, por lo tanto, se centra la investigación en diferentes algoritmos y modelos que lo permiten.

Se han elegido tres modelos se basan en las copulas gausianas, redes generativas adversarias y redes bayesianas, cada uno. Se generan los datos y se estudia cómo de $buenos$ son.

La evaluación de estas se lleva a cabo por dos lados. Por uno, mediante una prueba de hipótesis, se evalúa si la distribución sintética es similar (estadísticamente) a la distribución original, comparando estadísticos como la varianza y la mediana. Por otro lado, se evalúa si se mantienen las correlaciones de los datos originales en las sintéticas mediante un análisis de similitud utilizando matrices de coeficientes de correlación. Se efectúa una diferencia de matrices (original y sintética), la cual, idealmente, debería ser muy cercana a cero.

Finalmente, se analizan los resultados de las dos evaluaciones y se proponen ideas para seguir esta rama de la investigación. Además, se analiza el impacto económico y ambiental del proyecto.

ABSTRACT

This final degree dissertation focuses on a very common problem nowadays: the lack of data. Data scientists need large volumes of data to produce accurate and precise tools.

In this document, research is carried out on a set of COVID patient data, which is described in all its aspects. The objective is to produce synthetic data, therefore, focusing the research on different algorithms and models that allow it.

Three algorithms have been chosen that are based on Gaussian copulas, adversarial generative networks, and Bayesian networks, each one. The data is generated and studied how $good$ it is.

The dataset is evaluated in two ways. On one hand, performing a hypothesis test, the synthetic distribution is evaluated to be similar (statistically) to the original distribution, comparing statistics such as variance and median. On the other hand, it is evaluated whether the correlations of the original data are maintained in the synthetic ones through a similarity analysis using correlation coefficient matrices. A difference of matrices (original and synthetic) is carried out, which should ideally be very close to zero.

To sum up, the results of the two evaluations are analyzed, and ideas are proposed to follow this branch of research. In addition, the economic and environmental impact of the project is analyzed.

GLOSSARY

APACHE_II: Acute Physiology and Chronic Health disease Classification System II

BMI: Body Mass Index

CNN: Convolutional Neural Network

CRP: C-Reactive Protein

CSV: Comma separated values

CTGAN: Conditional Generative Adversarial Network

DBP: Diastolic Blood Pressure

DD: DataDescriber

DG: DataGenerator

DS: DataSynthesizer

FiO2: Fractional Inspired Oxygen

GAN: Generative Adversarial Network

GC: Gaussian Copula

GCS: Glasgow Coma Scale

ICT: Information and Communication Technologies

IT: Information Technology

MWU: Mann-Whitney-Wilcoxon U Test

PaCO2: Partial Pressure of CO2

PaO2: Oxygen partial pressure

PPE: Personal Protection Equipment

ROX: Ratio of Oxigen

SBP: Systolic Blood Pressure

SDV: Synthetic Data Vault

SMOTE: Synthetic Minority Oversampling Technique

SOFA: Sequential Organ Failure Assessment

SpO2: Oxygen Saturation

Acknowledgments

Albert Einstein, a while ago, said ”It’s not that I’m so smart, it’s just that I stay with problems longer.” I wouldn’t dare to compare myself to him, but definitely this is something that represents me. This thesis has been a complete challenge for me, Data Science is a wonderful discipline with a lot of different fields, each one wider than the one before. Besides from wide, this fields are complex, they involve maths, statistics, computer science, programming,... and much more knowledge and skills needed to succeed. So, I really had to stay stuck to the problem. There’s been a lot of up and downs, big goals, step backs that needed to be taken, and the road was everything but plain. Anytime anyone encounters this kind of roads, the worst thing you can do is drive it by yourself and luckily my journey’s been filled with amazing people that I want to give special thanks in this space.

First of all, thanks Dra. Magda L. Ruiz and Dr. Luis E. Mujica, for trusting me and seeing potential from day one. I know this is just a little part of what we wanted to do, and I really hope you can get there eventually, every idea you have is amazing and I’m a hundred percent sure that you will. Thanks for keeping an eye on me these months where I had to be a very multitasking person, in a moment in my life full of changes and important decisions, for giving me the space I needed and for being so uplifting in every meeting we had, this is what really got me through the semester. You always provide such valuable insights and without them, this $output$ would have been much different (I hope no one ever knows what different means!).

My friends, all of them, to all the people that have been with me during this time, it doesn’t matter if you’ve been with me the whole degree, the whole thesis, just the last month or a whole life, thank you for understanding me every time I had to say ”No, I can’t, I have to work on my dissertation”, thank you for staying. Specially I want to thank Carlos Pozo, I already consider you a friend, not just an ex-teacher, for always helping me out when I needed it, for bringing sincerity and motivations in each chat we had. Also Cristina Mendoza, you’ve been probably my alter ego during the last year and a half and I just can’t stop admiring you and feeling so happy for what life will bring you in your next steps, thanks for being there always. And of course, to all my friends in Castelldefels, my day to day, my past, my present and hopefully my future, thanks for being my safe place when everything seemed so dark and greedy, by your side everything looked brighter!

And last but not least, my parents and siblings, for being that rock you sit down to catch your breath, I found it in every step of the way. Thank you for always supporting my decisions, for listening my overthinking about every single word in this dissertation, for faking your understanding (about the topic) so I would just easily express what I needed to. You’ve been my diesel, thank you.

Contents

List of Figures

List of Tables

1 INTRODUCTION

Two years ago from now, the outbreak of Covid-19 pushed the world to a lot of situations it had never faced before. All the faces of our society experienced a crisis: older and younger people, economy, politics,... But outstanding, the healthcare system experienced a big crush. Hospitals and healthcare centers were crowded of diseased people, the shortage of ventilators, PPEs, human resources... made very clear how fragile was the whole system and its failure had several consequences. Of course, the pandemic brought a lot of serious outcomes, from which loads of people, countries and economies are still recovering. Nonetheless, there's one result of the whole system crisis that had a massive impact among the several branches of our society: the lack of data.

The world's response to coronavirus faced data gaps, inadequate and inconsistent definitions of data across different governmental jurisdictions, ambiguous timing in reporting, problems in accessing data, and changing interpretations from scientific institutions[16]. This not only represented a problem for decision making during the epidemiological situation, but also is supposing now a problem for data scientists who are trying to originate resources for healthcare systems.

Therefore, a bunch of new questions emerge such as, how can more data be obtained? Are these processes valid? What can be done once the amount of data starts to be consistent? All these questions are transformed into goals and are studied along the report.

1.1 Dissertation Goals

As it's been mentioned before, a lot of sectors in now-a-days world are still dealing with data's undersupply, specially the public health one, mostly due to a lack of IT resources, consensus in metrics and reporting and privacy issues. The aim of this research is to study possible ways to solve this matter and get through another obstacle, so more tools can be developed and implemented in hospitals and healthcare centers. Specifically, in this bachelor thesis it's expected to:

1. Generate synthetic data using different algorithms.

2. Evaluate how similar is the synthetic variable compared to the original one.

3. Evaluate if the synthetic dataset preserves the correlations from the original one.

1.2 Nomenclature

Since a lot of specific words appear repeatedly throughout this document, in this space the standard nomenclature for most of them will be defined (abbreviations can be found in the Glossary).

Firstly let's split data into original and synthetic. Original data is the dataset provided by the school and where the works rely on. Synthetic data is all those numeric values that are the output of a model. Each dataset is constructed of m columns, which are called **variables**, and n rows, which are called **observations**. Also, it's important to mark the difference between new data and $synthetic$ data, the first is only used in Data Generation section and scripts, while the second one is used for everything else in this document. Other symbols and specific names are explained along the report.

2 Script set up

This thesis uses the code that can be found in the Appendix, here is explained how to reproduce this results from scratch using the provided code. Before you start make sure to work in only one space, this is key in order to respect the dependencies between code and files (most of them are local).

2.1 Directories

Once its chosen the workspace, its needed to create the following directories in it:

1. Gauss_Cop_Dist
2. GAN_Dist
3. DS_Dist
4. Dist

Each of them will store the executed code results.

2.2 Executing the script

2.2.1 Jupyter Notebook

Find the Jupyter Notebook code and reproduce it in a notebook. It's important to change the $path$ variable according where the files are or they need to be. This code executes the synthetic data for DataSynthesizer.

2.2.2 PyCharm

This bachelor thesis has been developed using PyCharm, any other script editor should work correctly. Now it's time to copy and paste the scripts and assignin the indicated name. The order of execution should be:

1. $data\_gen.py$: prepares the dataset for its later usage.
2. $dist\_plots.py$: plots all of the variables distribution and saves them in the Dist directory created earlier.
3. $algorithm\_1.py$: stores all of the necessary functions that the different models will need to produce the desired output.

4. $gc\_gen.py$: Gaussian Copula, results stored in its directory.

5. $gan\_gen.py$: Generative Adversarial Network, results stored in its directory

6. $ds\_gen.py$: DataSynthesizer, results stored in its directory.

Each directory, except *Dist*, should end up having 3 other directories (24 plots), 3 correlation matrix, 9 CSV files (hypothesis, correlation matrix and metrics comparison), 3 heatmaps, a 3x1 heatmap figure and a 6x4 distributions figure.

3 Data Generation

Data generation is an aggregation of machine learning techniques that have been created with the purpose of having a bigger datasets, going from $n_{original}$ observations to n_{new} observations where, generally, the new dataset is bigger than the original dataset. There are several methods in this field and they are used depending on the purpose of the generation, even though they all follow a similar structure when it comes to generate data, actually, it's the typical workflow for almost all data science projects, see the picture below.

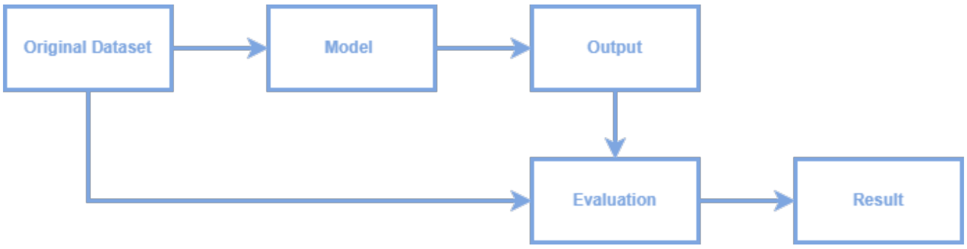


Figure 1: Typical and simplified workflow for generating data

3.1 Resampling

When it comes to model training, chances are that, when a lot of variables are involved and all all training data is used, the model becomes over-fitted . In order to avoid this and obtain more information on the performance of a model, the best tool is resampling. Even though it's not a data generation technique itself, it's important to know how this helps increasing the performance metrics of a model. It basically consists of repeteadly drawing samples from the training data and refitting the model, this can help the algorithm find any underlying pattern that maybe it haven't taken into account previously. There are two common ways of doing resampling[1]:

1. **Cross Validation**. This process evaluates the model on different subsets of data and later on use the other subsets to check what did the model learn. It's commonly known as splitting data, resulting into training and test data, of course, this happens repeatedly in order to produce the resampling. There different techniques such as K-Fold, Leave-One-Out, Hold Out.

2. **Boot-strapping**. Bootstrapping is a statistical technique for estimating the sampling distribution of an estimator (statistical parameter, i.e. mean, median,...) by generating random samples from the original data and computing the estimator for each sample. It is often used to

estimate the standard error of an estimator, or to select a sample of a certain size to achieve a desired level of precision.

3.2 Imbalanced datasets

A lot of times, data scientists encounter themselves in a situation where there's more data of a class A than from a class B (in this case there's much more information about male patients (A) than women patients(B)). This can lead to a biased algorithm and producing, therefore, biased conclusions. SMOTE (Synthetic Minority Oversampling Technique) [7] faces this matter, it can be done by over sampling the minority class or undersampling the majority class.

1. **Undersampling**: consists of discarding a number of observations of the class that is presented too often. This is a very straightforward technique, but it's only recommended to use when there's a big amount of data, since the scientist would need to end up having still a lot of observations to develop an accurate tool.

2. **Oversampling**: intuitively, it's the opposite of undersampling, add more data to the class that is presented less often. Typically the existing data is duplicated ending up with more observations of this smaller class. The problem here is that just plain oversampling won't add new information to the dataset, usually because it's done using duplicated observations.

The great thing of SMOTE is that generates data with a small perturbation that leads to complete new data. This happens because the algorithm gets an observation as an input, performs k-nearest neighbors, takes a vector between the observation and a neighbor, multiplies the vector by a number between 0 and 1, adds the vector to the dataset and the new data point is already done. This is a simple explanations of how SMOTE works, for more clarity, the diagram in Figure 2 has been made.

3.3 Synthetic Data

There's also the case when you do not need to resample your data in order to obtain a better performance of a model, duplicate values or synthesize news for dealing with imbalanced datasets, sometimes you just need to expand your source in order to have more observations and being able to develop better tools. The point is removing the barrier of data lack and this section is the core of the whole project.

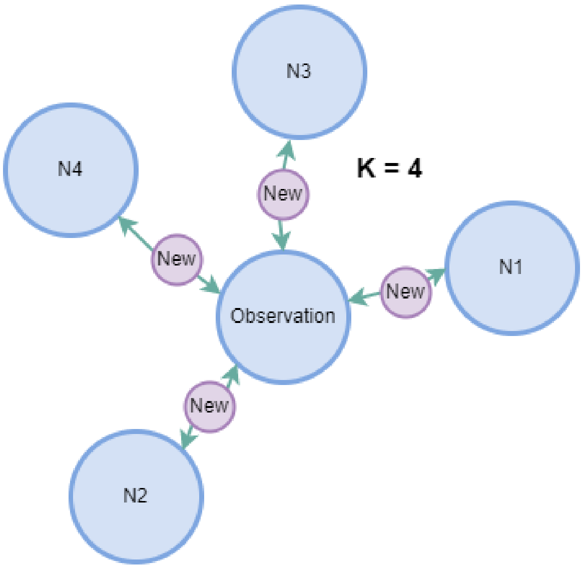


Figure 2: Simple SMOTE visualization. Uses K-Neighbors, being K=4.

Synthetic data is completely new data generated by an algorithm. It's needed to be similar statistically speaking (the variables themselves and the relationships between them). In the following sections this is going to be uncovered and studied with further detail.

4 Data Description

The dataset this dissertation works on is provided by *Hospital Clinic de Barcelona*. The aim of this section is to briefly understand which variables the models will have to take into account.

4.1 Data Structure

Cambridge Dictionary provides this definition for dataset: *A dataset is a collection of separate sets of information that is treated as a single unit by a computer*. In this case, the dataset is tabular and its composed of 27 variables and 156 observations. It hasn't been the need for cleaning or processing the data since it already was ready for working on it, the models neither specify any kind of preprocessing for the input. Being this said, it's important to clarify that the following variables won't be taken into account for any process in this dissertation:

- $Unnamed : 0$. This variable is the dataset index, it does not add any valuable information and, therefore, not needed during the whole project.
- $Hosp\_ID$. It's the patients identification, for privacy reasons this variable is not used in this project.
- $APACHE\_II.factor$ This is the same as $APACHE\_II$, but factorized, so it's a discrete version. The factors do not add any kind of valuable information and it makes the exercise much more complex, that's why it's been also removed.

4.2 Variables

Now let's study the variables presented in the data. Demography, meaning and distributions are the main points in the description.

4.2.1 Meaning

In Table 1 each variable is accompanied of a brief description of its meaning and which type of data is it.

Variable	Definition	Data Type
Age	Patient's age	Integer
Gender	Patient's gender	Integer
Heart_rate	Patient's heart rate	Integer
SBP	Systolic Blood Pressure	Integer
DBP	Diastolic Blood Pressure	Integer
SpO2	Oxygen saturation	Integer
Max_SpO2	Maximum oxygen saturation	Integer
Temperature	Patient's temperature	Float
BMI	Body Mass Index	Integer
pH	Acidity measure	Float
PaCO2	Partial pressure of CO_2	Float
GCS	Scoring system to measure patient level of consciousness	Integer
APACHE_II	Disease severity classification system used in the ICU	Float
SOFA	Sequential Organ Failure Assessment	Integer
PaO2_FiO2	Ratio of arterial oxygen partial pressure to fractional inpired oxigen	Integer
Creatinine	Byproduct left over during the metabolism stage in muscle proteins	Float
Leucocytes	Type of blood cells made in the bone marrow that is part of the body immune system	Float
CRP	Protein synthesized and released by the liver when there's inflammation in the patient's body	Float
D_dimer	Protein fragment made after a blood clot dissolves	Integer
ROX_index	Ratio of oxygen saturation by fraction inspired oxygen	Float
Symptom_to_ICU	Coeficient	Integer
Hosp_to_ICU	Coeficient	Integer
Survival	1 if the patient has survived and 0 if the patient did not	Integer
Intubation	1 if the patient has been intubated and 0 if the patient hasn't	Integer

Table 1: Variables description: definition and type of data

4.2.2 Descriptive Statistics

Now the description takes a more statistical approach. For the statistical description it's been chosen two position and a dispersion metrics, together with the minimum and maximum of each variable. This is shown in Table 2

Variables	Mean	Median	Variance	Minimum	Maximums
Age	62	62	134,56	30	90
Gender	0,28	0	0,2	0	1
Heart_rate	81,41	80	277,06	35	140
SBP	122,84	121,5	382,82	68	180
DBP	67,39	67	155,69	30	110
SpO2	89,73	90	32,17	60	99
Max_SpO2	96,76	98	9,7	86	100
Temperature	36,88	36,8	0,97	34,5	39,7
BMI	28,04	27	24,47	17	46
pH	7,43	7,45	0,01	7,2	7,59
PaCO2	35,53	33	82,92	16,9	71
GCS	14,75	15	1,95	3	15
APACHE_II	11,71	11	25,28	2	26
SOFA	4,92	4	5,41	0	13
PaO2_FiO2	133,38	124	4059,31	42	395
Creatinine	1,02	0,84	0,46	0,37	6,23
Leucocytes	8,06	7,12	20,1	1,24	27
CRP	52,4	16,39	7536,29	0,4	500
D_dimer	2252,88	900	21771405,62	200	36200
ROX_index	6,36	5,71	14,04	2,12	30,64
Symptom_to_ICU	10,26	9	35,79	1	45
Hosp_to_ICU	2,61	2	14,12	0	34
Survival	0,85	1	0,13	0	1
Intubation	0,5	0,5	0,25	0	1

Table 2: Statistical description of the variables

Furthermore, for a clearer understanding of the population, in Figure 3, it's plotted each variable's distributions as density histograms.

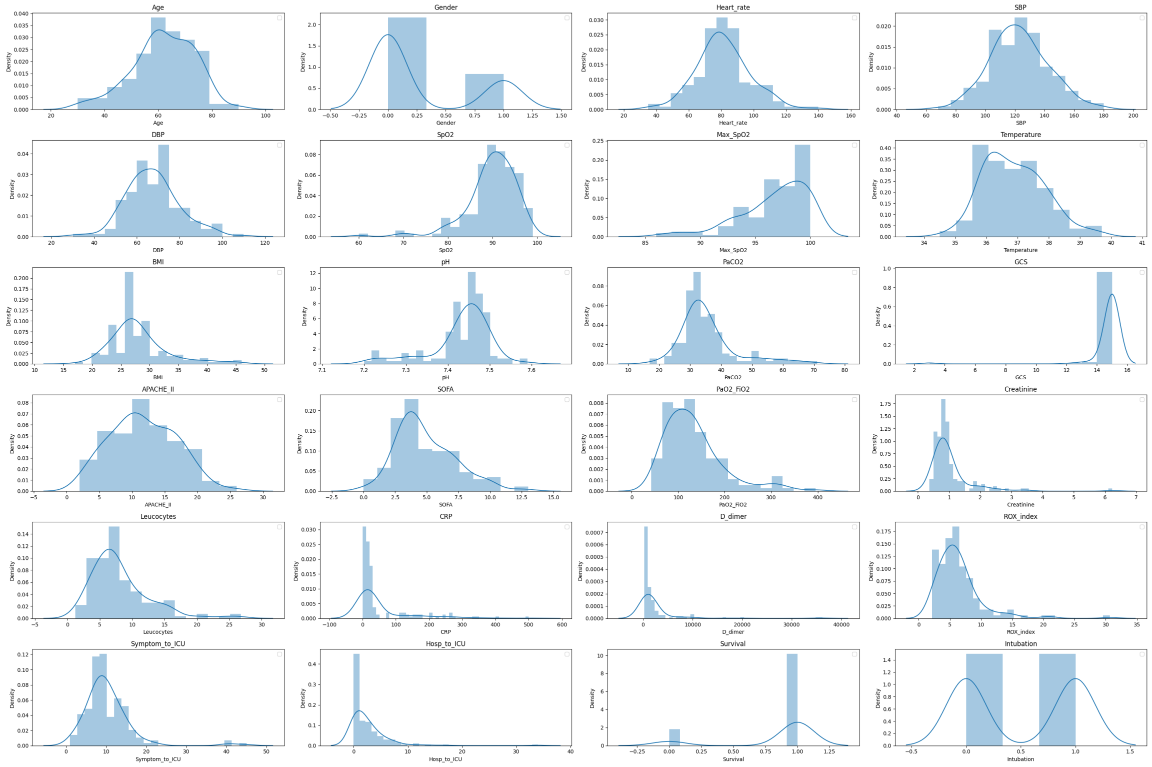


Figure 3: Density histograms for each variable in the dataset.

4.3 Limitations

As it's been said previously, the amount of data is decisive when it comes to train a model. In this case, there are only 156 observations. Definitely this is not a trustworthy amount of data [5] to develop any kind of regression or classification model, because the relationships between medical variables is not usually straightforward to see, a very complex pattern lays underneath these values. Even though there isn't a set threshold for having enough data, understanding the purpose (health-care system tool) and the background (type of variables and number of observations) allows understanding that having barely 150 observation won't make it. In the following chapter three models for generating data are explored and tested.

5 Data Generation

In previous sections the dataset it's been described briefly in order to understand which types of variables are going to be worked with. This is now the state of the art of this project, analyzing how more data can be generated and how usable is it. Here it can be found the explanation for each chosen algorithm, the code implementation for this can be found in the Appendix.

5.1 Algorithms

The presented algorithms follow the same structure across the project. First data is generated for each algorithm and three different cases of synthetic data: 50, 100 and 150 synthesized observations. Once data is generated then its plotted and compared to the original distributions. As a final step, there is an evaluation of this data using two different methods (see Data Evaluation section). The workflow in Figure 4 summarizes the whole process.

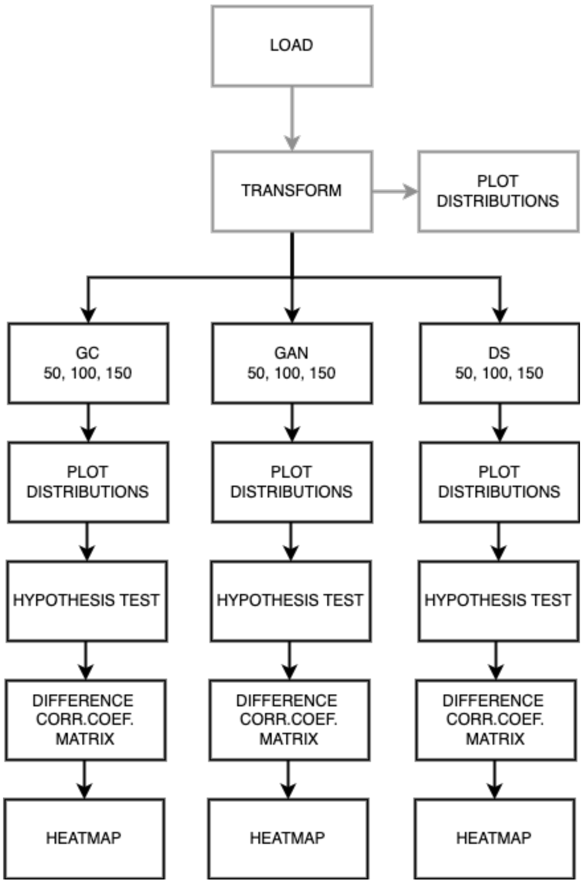


Figure 4: Flow diagram for the whole project's code.

For this project, the following algorithms have been chosen to generate synthetic data and evaluate their output. The selection criteria it's bibliographic, there is no information on synthetic patients data itself, so it's been researched with methods are used for a more general issue, and then applied to the thesis problem. Amongst several algorithms, for its accessibility, usability and reliability, the ones that fitted the best for this thesis are (taking into account bibliographic and time limitations):

1. Gaussian Copula (GC)

2. Generative Adversarial Networks (GAN)

3. DataSynthesizer (DS)

We'll deep dive into them in further sections, but firstly, let's start with the common steps. Firstly a general transformation happens in the dataset, where it's loaded and some columns are deleted. It's important to keep in mind that for generating ideal additional data two conditions must be fulfilled:

1. The new data variables should individually behave exactly as the original ones.

2. The new data should keep all the dependencies across variables and dimensions.

All code-related explanations can be found in the Appendix, in this section only results and models fundamentals will be studied in depth.

5.2 Gaussian Copula

Copulas (*statistical models*) are actually a tool for attacking this two issues This is not easy to achieve, but the goal lies in getting as closer as possible to these to conditions, therefore ending up with a new dataset that behaviour and dependencies are the most similar possible to the original dataset. Let's give it a glimpse from the mathematical side and then a definition more oriented to the matter itself [22].

In probability theory and statistics, a copula is a probability multivariate distribution function whose marginal distributions for each variable are uniformly distributed. Copulas describe the dependencies' structures between random variables. Its mathematical definition: A *n*-dimensional copula or *n*-copula is a function $C : [0,1]^n \to [0,1]$, this means that the copula takes as input d variables that are each uniformly distributed on the interval $[0,1]$, and maps them to d variables that are also uniformly distributed on the interval $[0,1]$. Also, a copula, fulfills the following conditions [9][8]:

- $C(1, ..., 1, u_k, 1, ...1) = u_k$ for each $k \leq n$ y $\forall u_k \in [0, 1]$, the copula is equal to u_k if any of the arguments is u and the rest is 1. Basically this means, that the copula will only represent the dependencies of variable u_k and the rest.
- $C(u_1, ..., u_{k-1}, 0_k, u_{k+1}, ..., u_n) = 0$ for any $k \leq n$, the copula is zero if any of the arguments is zero.
- The copula is n-non-decreasing , meaning if one variable's value increases, the joint distribution of the remaining variables also increases.

How can this be used for generating data? Well, even though it's not the aim of this thesis understanding in deepness how generating data with copulas work, let's try to understand a bit more about copulas. Suppose its aimed to generate synthetic data from a probabilistic model for n variables $X_1, ..., X_n$. To achieve the first aim its needed to find appropriate marginal distributions $F_1, ..., F_n$. A simple approach is to approximate them by the corresponding empirical distribution functions. To achieve the second aim, however, its required to build a model for the joint distribution function $H(x_1, ..., x_n)$. The key result, Sklar's theorem [4], states that any joint distribution function can be written as:

$$H(x_1, ..., x_n) = C(F_1(x_1), ..., F_n(x_n)) \tag{1}$$

It basically says that any multivariate distribution can be represented as a combination of its marginal distributions ($F(x)$) and a copula function (C), which describes the dependence structure between the variables. [22]. Using this dependence structure described (usually using a correlation matrix) and reversing the building steps of a copula, synthetic data can be generated.

5.2.1 Data generation

For generating new data using GC, SDV python package [10] has been used. The Synthetic Data Vault (SDV) is a synthetic data generation ecosystem of libraries that allows users to easily learn single-table, multi-table and time series datasets to later on generate synthetic data that has the same format and statistical properties as the original dataset[25].

The basic idea behind using a Gaussian copula in SDV is to first standardize the original data so that all variables have a mean of zero and a standard deviation of one. Then, the correlation matrix of the standardized data is used to define the Gaussian copula. The Gaussian copula is then used to generate synthetic data by sampling from the multivariate normal distribution defined by the correlation matrix.

Once the synthetic data is generated, it is then transformed back to the original scale of the data. This synthetic data should preserve the same statistical properties as the original data.

In Figure 5 each variable's distribution is plotted and compared original versus synthetic. This case shows graphics for 50 new synthetic data (see the Appendix plot section for the two other cases).

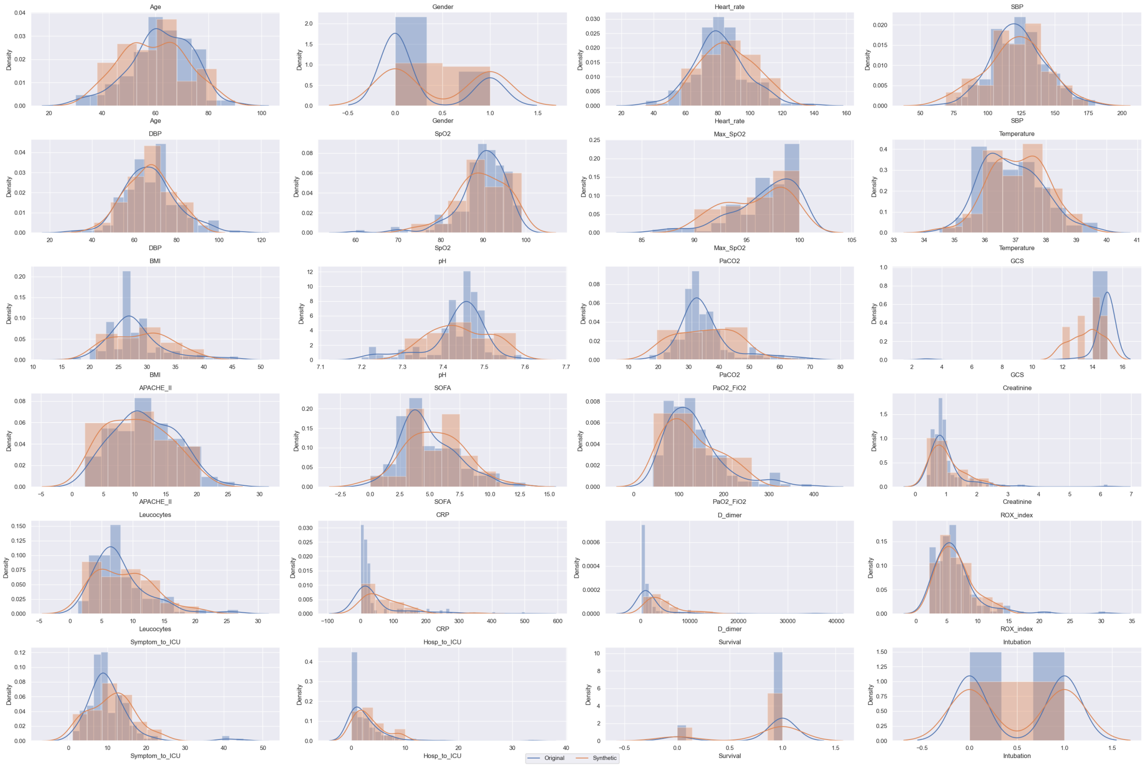


Figure 5: Distribution plots for each variable.

5.3 Generative Adversarial Network

GAN are algorithms that generates a completely new output. This kind of algorithms are frequently used in image generation, even though, in this thesis it's been intended to see how a GAN behaves when it comes to tabular data. Now let's see what are GAN, starting by understanding generative and discriminative models, followed by seeing these networks structures.

5.3.1 Generative Models

Before GAN are properly introduced let's make clear what a generative model is. Generative models are those which generates new data, in contrast to discriminative models which divides data depending on a label. Let's suppose there are data observations X and their labels Y [32].

- Generative models take into account the joint probability $P(X,Y)$ or $P(X)$ in case there are no labels.
- Discriminative models take into account the conditional probability $P(Y|X)$.

Generative models are way more complex than discriminative models since they need to understand all kind of relationships between data in order to produce a convincent output, while discriminative models are enough if they find a pattern that allows them to classify a data instance X into a label Y. Figure 6 explains very visually what this means:

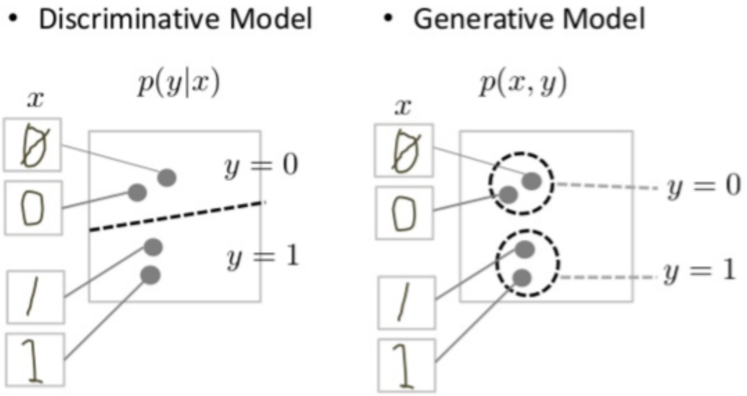


Figure 6: Difference between Generative and Discriminative Models [11]

To make it clearer,Figure 6, the discriminative models only needs to make a line to separate those instances that could be a 0 from those that could be a 1, while, as it can be seen, the generative models need to go over the whole distribution's space so they can reproduce an instance that's extremely close to the real one.

5.3.2 Structure of a GAN

A GAN is divided into two main parts and each of them is a neural network.:

- The **generator** learns to produce new possible data.
- The **discriminator** learns to distinguish fake data, generator's output, from real data.

When the process has just started is easy for the discriminator to learn that the generator's data is fake, when this happens generator is penalized using back-propagation (gradient descent algorithm in neural network) so it has to adjust network's weights in order to produce a more convincing output. As the iterations go by, the discriminator will encounter more difficulties to discern which data is fake and which is original. The algorithm is over once the discriminator can't distinguish between real and fake data.

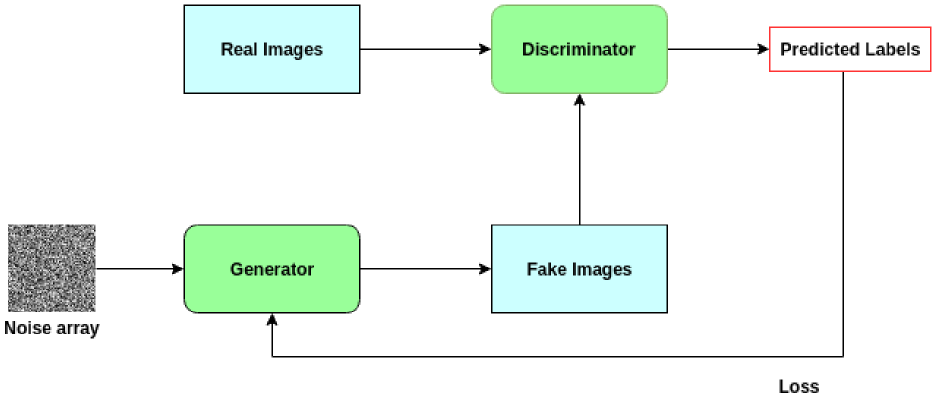


Figure 7: GAN's structure diagram[28].

Figure 7 simplifies the structure of generative adversarial network, where the loss is the penalization it's been mentioned earlier.

5.3.3 Data Generation

For the algorithm, again, SDV library has been used, but this time using CTGAN model. CTGAN (Conditional Generative Adversarial Network) [37] is a type of generative model that uses a combination of deep learning techniques and adversarial training to generate synthetic data that is similar to real data. It is particularly useful for generating synthetic data for use in data privacy and data augmentation.

CTGAN model works as any other model of generative adversarial network, a generator, a discriminator and a loss function that is a regulator for making data, each iteration, more similar to the original one.

One of the key features of CTGAN is its ability to condition the generation of synthetic data on specific features or variables of the data. This allows the user to control the characteristics of the synthetic data and make it more closely resemble the real data in specific ways. Nonetheless, this is beyond the scope of this project and the CTGAN model is not modified in any situation.

In Figure 8 each variable's distribution is plotted and compared original versus synthetic. This case shows graphics for 50 synthesized observations (see the Appendix plot section for the two other cases).

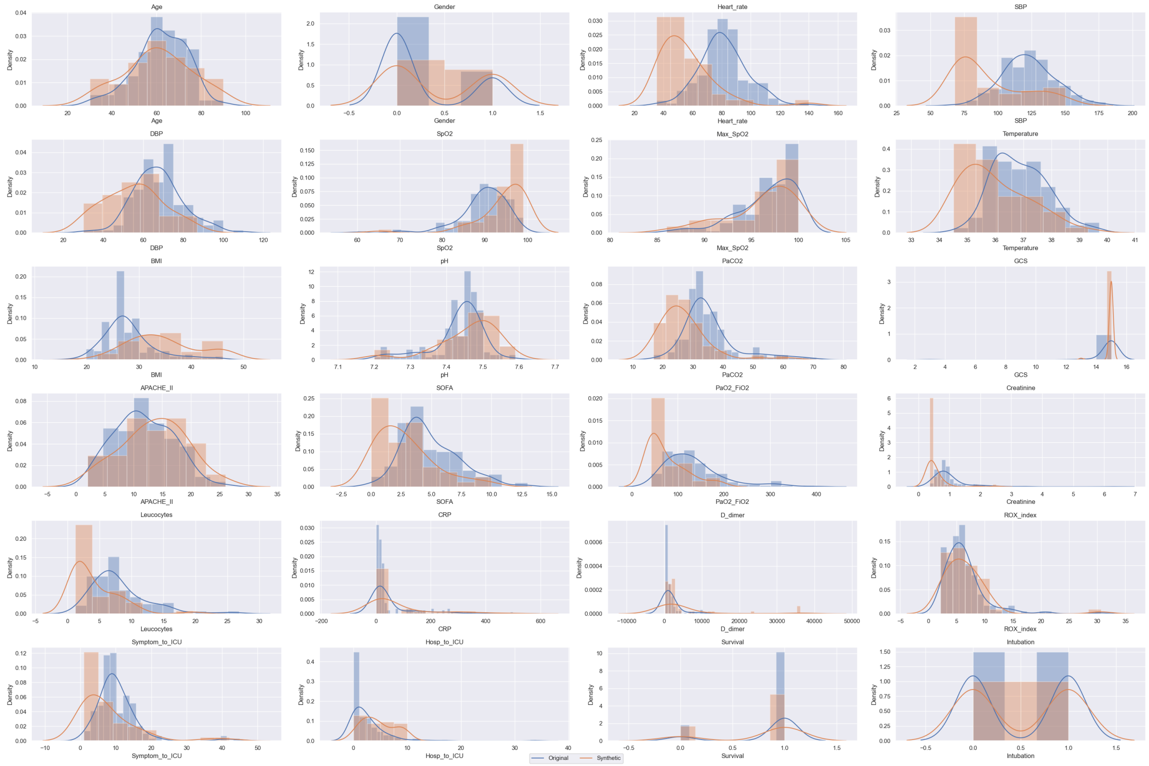


Figure 8: Distribution plots for each variable.

5.4 DataSynthesizer

DS [26] is a tool developed by the University of Washington Information School around 2017. Since data governance is a very little explored field in data science, all the legal processes of sharing data between social scientists, government agencies, health workers and data scientists are slow and expensive. The report sets the typical time frame of these processes at 18 months. Researchers found here an opportunity to accelerate the bridges across those organizations and data scientists by giving them data similar enough to work on anything they intend to but taking into account:

1. Anonymize the data and guarantee security and privacy

2. Preserve the original data characteristics so the tools developed have the desired output.

The aim of this tool is to produce a dataset that is statistically close to the original one. First, let's understand a bit the fundamentals of Bayesian Networks and how DS uses them to generate new data.

5.4.1 Bayesian Network

A Bayesian network is a graphical model[34] that represents the probabilistic relationships between different variables. Each node in the graph represents a random variable, and the edges between nodes represent dependencies or relationships between the variables[19]. The probability of each variable is computed using Bayes' theorem, which allows reasoning about the uncertain events based on the known probabilities of related events.

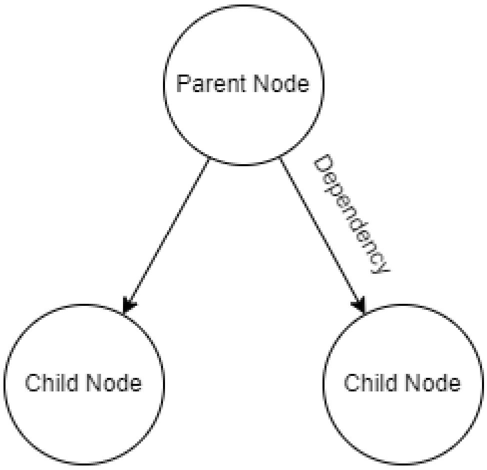


Figure 9: Simple schema of node relationships.

For example, a Bayesian network could be used to model the probability of a person having a certain disease based on their symptoms and other risk factors. The network would include nodes for each of these variables, and the edges between the nodes would represent the relationships between them. Using the probabilities associated with each node and the dependencies represented by the edges, we can make predictions about the probability of a person having the disease given certain symptoms or risk factors.

Overall, Bayesian networks provide a flexible and powerful way to reason about uncertain events, and are widely used in a variety of applications, including decision making, risk assessment, and machine learning.

5.4.2 Data Generation

DS is made out of three modules:

1. DataDescriber: Investigates the data types, correlations and distributions of the variables in the original dataset. It produces a data summary and it can add noise in order to preserve privacy.

2. DataGenerator: samples from the summary computed by DataDescriber (DD) and outputs synthetic data

3. ModelInspector: shows an intuitive description of the data summary that was computed by DataDescriber, allowing the evaluation of the summarization process accuracy.

In this case, DS operates in *correlated attribute mode* (it also operates in two other modes, but they are not studied here. See the reference for more information). This mode is based on a Bayesian Network capturing the correlation structure between attributes, then it draws samples from this model to construct the dataset.

The whole process structure looks like workflow found in Figure 11

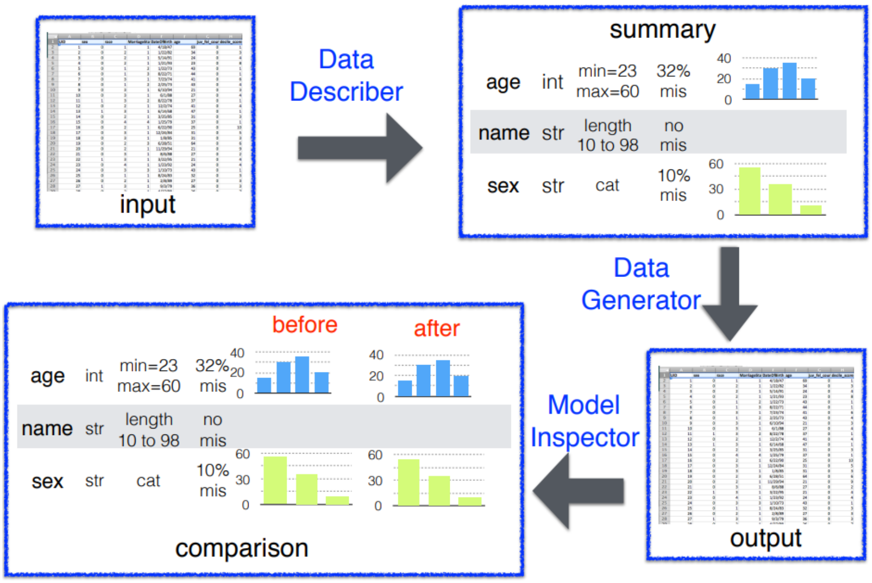


Figure 10: DataSynthesizer Process[26].

DataDescriber

In this case, it is intended to capture any kind of correlation that may exist between all of the dataset variables, that's why we use *correlated attribute mode*. DD uses the GreedyBayes algorithm to construct bayesian networks to model correlated attributes. This algorithm takes as a function attributes a dataset, a set of variables and the maximum number of parents and returns a bayesian network. This network constructed by Greedy Bayes, gives the sampling order for generating variable values. The code implementation for this section can be found in Appendix.

DataGenerator

Given a dataset description file generated by DD, DG samples synthetic data from the distributions in this file. The size of the output dataset is specified by the user, and defaults to n, the size of the input dataset. DG algorithm generates the data by the order set in the DD previously. The code implementation can be found in Appendix). In Figure 11 each variable's distribution is plotted and compared original versus synthetic. This case shows graphics for 50 new synthetic data (see the Appendix plot section for the two other cases).

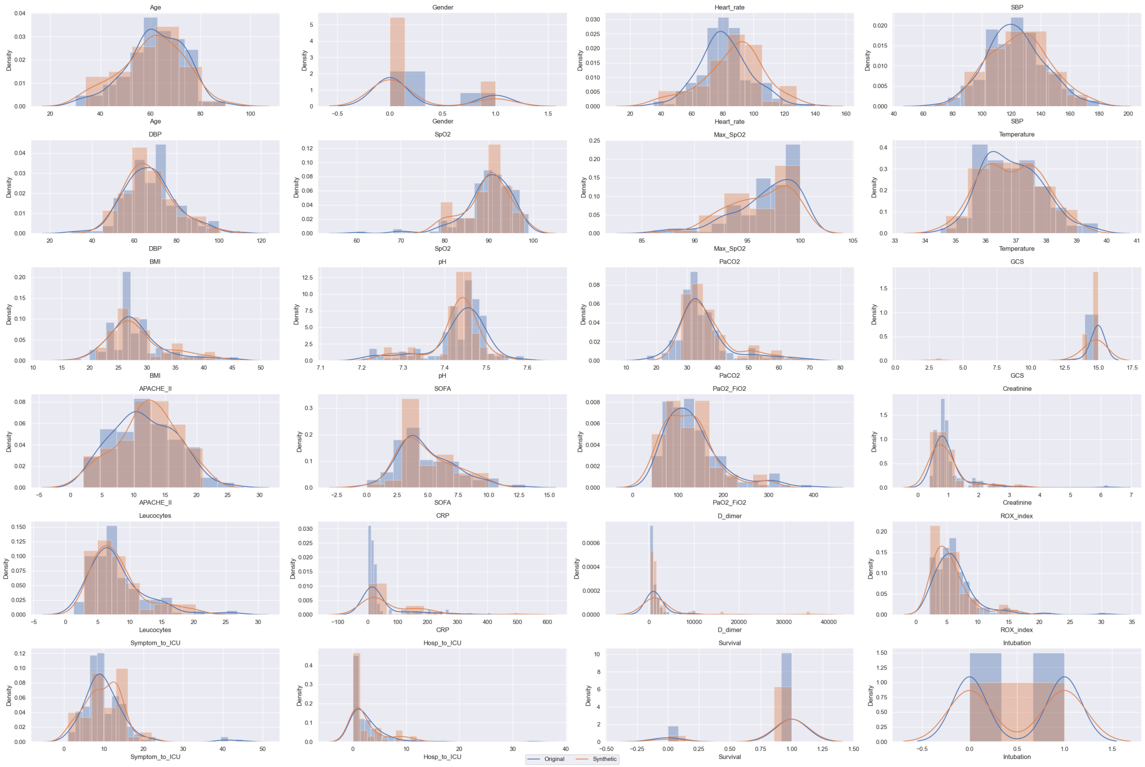


Figure 11: Distribution plots for each variable.

6 Data Evaluation

Given that the original distribution needs to be reproduced, it'd be expected that, statistically, original and synthetic one, are as similar as possible. Let's recall that there are two goals that need to be accomplished as its been seen previously: to keep the individual behaviour of each variable and to keep the dependencies across variables. In this dissertation hypothesis test and correlation matrix similarity analysis are used to evaluate those two aims.

6.1 Hypothesis Test

This section allows understanding how close are the synthetic distributions to the original ones. In statistics, the most common test for evaluating hypothesis is t-test. T-test is a parametric statistical test that contrasts the null hypothesis (H_0) of two population's mean being the same, against the alternative hypothesis (H_1) of not being the same[2]. Original variable's mean should be equal to synthetic variable's mean, as null hypothesis, otherwise, would be the alternative:

$$H_0 : \mu_O = \mu_N \tag{2}$$

$$H_1 : \mu_O \neq \mu_N \tag{3}$$

Even though t-test's simplicity, the following conditions must be fulfilled by the dataset:

- Variables need to be **independent**
- Variable's distribution must be **normal**
- **Variances** must be the same in each distribution (homoscedasticity)

Unfortunately, the second condition is not achieved by the used data for this thesis. In order to contrast hypothesis, the **U-Test** or **Wilcoxon-Mann-Whitney Test** needs to be performed.

6.1.1 U-Test

U-Test or Wilcoxon-Mann-Whitney Test is a non parametric statistical test that contrasts if two samples are from equally distributed populations. If two compared samples are from the same population, when data is put together and sorted ascendant, it would be expected that data from one and other sample would be intercalated randomly. Otherwise, if one of the samples belongs to a population with greater or smaller values than the other population, once the data is sorted, these will tend to group together in a way that one sample's data stays over the other. This is the idea which the U-Test is based on. Accordingly to this idea, the test contrasts that the probability of an observation of population 1 is on top of an observation of population 2 is the same:

$$H_0 : P(x > y) = P(y > x) \tag{4}$$

$$H_0 : P(x > y) = 0.5 \tag{5}$$

$$H_1 : P(x > y) \neq P(y > x) \tag{6}$$

$$H_1 : P(x > y) \neq 0.5 \tag{7}$$

In this case, H_0 and H_1 would be:

- H_0: The probability of an observation x from the original dataset of being right on top of an observation y from the synthetic dataset is the same.
- H_1: The probability of an observation x from the original dataset of being right on top of an observation y from the synthetic dataset is not the same.

The median it's a widely used parameter for this test, although this is only valid when position is the only distribution's feature that's not the same for both, original and synthetic. U-Test is less powerful, statistically speaking, but it's more robust than t-test. There are some conditions to fulfill as well:

1. Variables need to be **independent**
2. Variable's distribution does not have to be **normal**. However, if the test compares medians, distributions from different populations must be the same (original dataset versus synthetic dataset)
3. Data must be able to be sorted in ascending order.
4. **Variances** must be the same in each distribution (homoscedasticity, see Levene's Test) section

For calculating the U statistic of the u-test the following formula's been used (previously, the two populations observations must have been merged, sorted and ranked):

$$U = \text{min}(U_1, U_2) \tag{8}$$

$$U_1 = n_1 \cdot n_2 + \frac{n_1(n_1+1)}{2} - R_1 \tag{9}$$

$$U_2 = n_1 \cdot n_2 + \frac{n_2(n_2+1)}{2} - R_2 \tag{10}$$

where,

n_1 is the number of observations in population 1,

n_2 is the number of observations in population 2,

R_1 is the rank sum of population 1,

R_2 is the rank sum of population 2,

The U statistic is used to calculate the p-value and then the hypothesis is accepted ($p < 0.05$) or rejected. This is done in Python (see code in Appendix) and the U statistic and p-value are calculated.

6.1.2 Levene's Test

The current data fulfills all of the conditions mentioned above but it's still unknown if the variances are the same for each variables' distribution. Checking this is easy using the Levene's test . It tests the null hypothesis that the variances of two populations are equal (homoscedasticity). If the resulting p-value of Levene's test is less than some significance level (typically 0.05), the obtained differences in sample variances are unlikely to have occurred based on random sampling from a population with equal variances. Thus, the null hypothesis of equal variances is rejected and it is concluded that there is a difference between the variances in the population[35]. The test statistic is called W and it's defined as:

$$W = \frac{(N-k)}{(k-1)} \cdot \frac{\sum_{i=1}^{k} N_i (Z_{i.} - Z_{..})^2}{\sum_{i=1}^{k} \sum_{j=1}^{N_i} (Z_{ij} - Z_{i.})^2} \tag{11}$$

and,

$$Z_{ij} = \begin{cases} |Y_{ij} - \bar{Y}_i| \\ |Y_{ij} - \tilde{Y}_i| \end{cases} \tag{12}$$

where,

k is the number of different groups to which the sampled cases belong,

N_i is the number of cases in the ith group,

N is the total number of cases in all groups,

Y_{ij} is the value of the measured variable for the jth case from the ith group,

$\bar{Y}_i$ is th mean of the i-th group,

$\tilde{Y}_i$ is a median of the i-th group,

$Z_{i.}$ is the mean of the Z_{ij} for group i,

$Z_{..}$ is the mean of all Z_{ij}.

For Z_{ij} case, both definitions are correct, the corresponding to the median one is also the definition for another test that won't be explored in this dissertation. Then the p-value is calculated using W statistic. This is done in Python (see code in Appendix) and the W statistic and p-value are calculated and those variables whose p-value is less than 0.05 will not be taken into account for the U-Test.

6.1.3 Results

The results of the whole analysis for three different cases (50, 100, 150) and three algorithms turns out in 9 results table. The extended tables can be found in the Appendix, where it's been organised in the Tables section, by algorithm. Table 3 is a summary of the extended versions, where the number of hypothesis accepted and rejected are counted. It shows the numbers for each case (50, 100, 150) and each algorithm.

New Data	Nº Hypothesis	GC	GAN	DS
50	Accepted	14	9	20
50	Rejected	10	15	4
100	Accepted	17	3	23
100	Rejected	7	21	1
150	Accepted	13	4	22
150	Rejected	11	20	2

Table 3: Results table for hypothesis test

6.2 Similarity Analysis

Hypothesis test let us know how similar were, for each variable, the original and the synthetic distribution. But this does not allow to see a bigger picture. It's very important also to understand if the relationships between variables, any possible dependency, have been kept in the synthetic data. In order to tackle this issue, the following procedure's been designed:

1. Generate the correlation matrix for the original dataset and the new one.

2. Subtract both matrix and it would be expected to have numbers as close as possible to zero.

3. Create a heatmap for making the results more visual.

4. Repeat for each number of new data created.

6.2.1 Correlation Matrix

In order to understand if the dependencies have been kept by the algorithms, it's necessary to keep in mind the idea of correlation. By definition, correlation measures how two or more variables are related to one and other [24]. Therefore, given a dataset of m columns and n rows, the correlation matrix, let it be R, will result as a $m \times m$ symmetric matrix with a unit diagonal. The R_{ij} elements are the Pearson Coefficients ρ between the element i and j, denoted as $\rho(i, j)$.

Pearson's correlation coefficient estimates the correlation of two variables, dividing the covariance by the product of each standard deviation coefficient:

$$\rho(m_i, m_j) = \frac{cov(m_i, m_j)}{\sigma(m_i)\sigma(m_j)} \tag{13}$$

where,

$$cov(m_i, m_j) = \sum_{k=1}^{n}(m_{i_k} - \bar{m_i})(m_{j_k} - \bar{m_j}) \tag{14}$$

$$\sigma(m_i) = \sqrt{\sum_{k=1}^{n}(m_{i_k} - \bar{m_i})^2} \tag{15}$$

$$\sigma(m_j) = \sqrt{\sum_{k=1}^{n}(m_{j_k} - \bar{m_j})^2} \tag{16}$$

therefore,

$$\rho(m_i, m_j) = \frac{\sum_{k=1}^{n}(m_{i_k} - \bar{m_i})(m_{j_k} - \bar{m_j})}{\sqrt{\sum_{k=1}^{n}(m_{i_k} - \bar{m_i})^2}\sqrt{\sum_{k=1}^{n}(m_{j_k} - \bar{m_j})^2}} \tag{17}$$

The following step is to generate each matrices (R_O, R_S) and subtract them,

$$R_O = \begin{bmatrix} \rho_{00_o} & \cdots & \rho_{i0_o} & \cdots & \rho_{m0_o} \\ \vdots & \ddots & \vdots & & \vdots \\ \rho_{0j_o} & \cdots & \rho_{ij_o} & \cdots & \rho_{mj_o} \\ \vdots & & \vdots & \ddots & \vdots \\ \rho_{0m_o} & \cdots & \rho_{im_o} & \cdots & \rho_{mm_o} \end{bmatrix} \tag{18}$$

$$R_S = \begin{bmatrix} \rho_{00_s} & \cdots & \rho_{i0_s} & \cdots & \rho_{m0_s} \\ \vdots & \ddots & \vdots & & \vdots \\ \rho_{0j_s} & \cdots & \rho_{ij_s} & \cdots & \rho_{mj_s} \\ \vdots & & \vdots & \ddots & \vdots \\ \rho_{0m_s} & \cdots & \rho_{im_s} & \cdots & \rho_{mm_s} \end{bmatrix} \tag{19}$$

Therefore, let's denote ΔC as the difference between both correlation coefficient matrices, C_O and C_S, as:

$$\Delta R = R_O - R_S = \begin{bmatrix} \rho_{00_o} - \rho_{00_s} & \cdots & \rho_{i0_o} - \rho_{i0_s} & \cdots & \rho_{m0_o} - \rho_{m0_s} \\ \vdots & \ddots & \vdots & & \vdots \\ \rho_{0j_o} - \rho_{0j_s} & \cdots & \rho_{ij_o} - \rho_{ij_s} & \cdots & \rho_{mj_o} - \rho_{mj_s} \\ \vdots & & \vdots & \ddots & \vdots \\ \rho_{0m_o} - \rho_{0m_s} & \cdots & \rho_{im_o} - \rho_{im_s} & \cdots & \rho_{mm_o} - \rho_{mm_s} \end{bmatrix} \tag{20}$$

This has been, as well the plots, computed using Python, the code can be found in the Appendix.

6.2.2 RSME

Root squared mean error is a widely used metric in data science field, since it's an accuracy measure for prediction models. For that kind of models, it's measured how close is a predicted value to the real one, so, the difference between both values is an error. RMSE serves to aggregate the magnitudes of the errors in predictions for various data points into a single measure of predictive power [36]. This applied to the similarity analysis, an analogy could be done: predicted observations would be synthetic data and real observations would be original data. RSME is computed using the following formulas:

$$RMSE = \sqrt{\frac{\sum_{i,j}^{m}(\Delta R_{ij})}{m}} \tag{21}$$

This is computed using Python (see the code in Appendix)

6.2.3 Results

Let's remember that the matrices are the difference between the original dataset correlation matrix and the synthetic one, then the heatmap is plotted. Firstly, there's a comparison amongst the same algorithm and its different number of synthesized data. The colors are adjusted in a numerical scale between values from 0 to 1, the darkest color is the maximum value amongst of all matrices, hence, the clearer the matrix, the higher similarity in correlation.

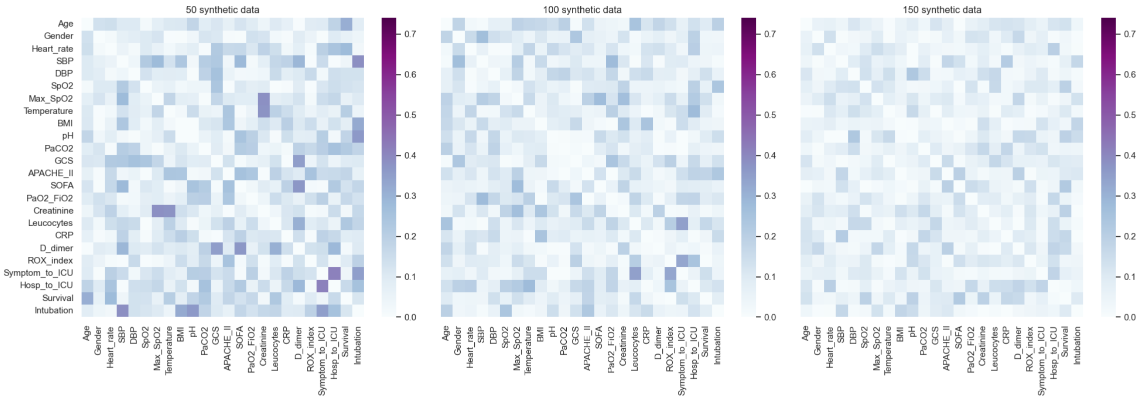


Figure 12: Heatmaps for GC algorithm.

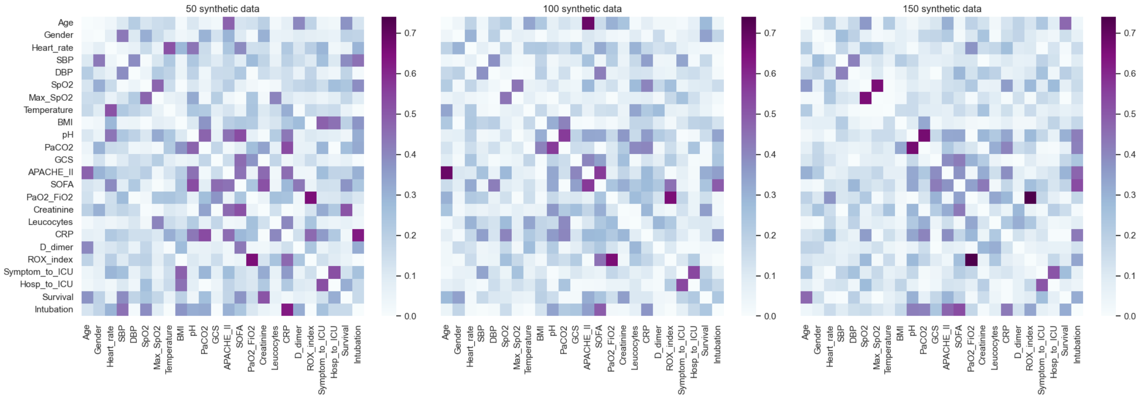


Figure 13: Heatmaps for GAN algorithm.

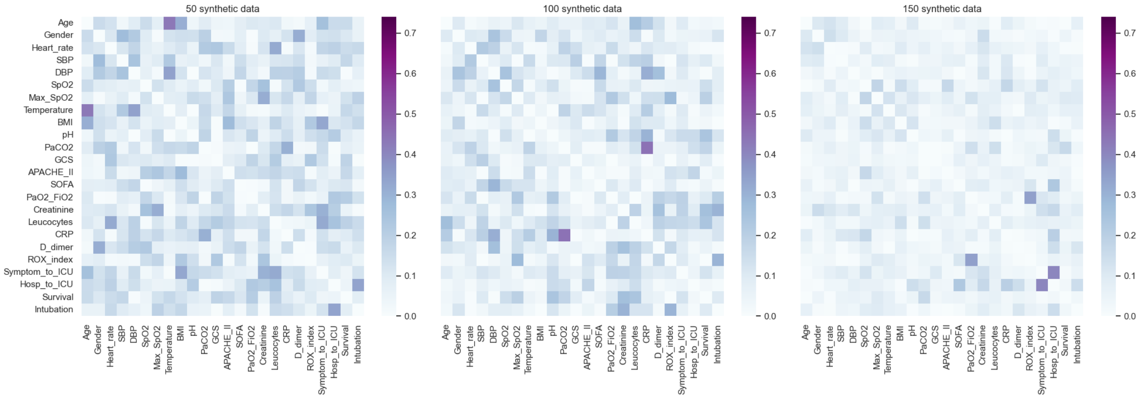


Figure 14: Heatmaps for DS algorithm.

In order to quantify the difference between all of them, it's been calculated the RSME for each algorithm and case. The results have been summarized in Table 4. As it can be observed, the DS is performing the best for almost all different numbers of new synthesized data, while GAN is the worst-performer.

New SD	GC	GAN	DS
50	0,140	0,221	0,134
100	0,112	0,189	0,113
150	0,096	0,179	0,073

Table 4: RSME Comparison for algorithm and number of synthetic data

7 Conclusions

The main goal of this dissertation is studying how feasible is to generate synthetic data for scientific usage. It's been explained how the lack of data limits researcher's work a lot of times, why this absence has happened during the pandemic years and which consequences is bringing at present.

Firstly, data's been processed and got rid off some variables that would complicate unnecessarily the workflow and not provide any useful information. Then it's been done a descriptive analysis of the dataset, a more medical one, where each variable is briefly explained, and a more statistical one, where position and dispersion metrics are put together in a table for each variable in the set. This, together with distribution plots, allows a general understanding of the data set and its size.

There are so many approaches for developing a project like this, but the one taken has been choosing three different algorithms, setting three different cases of synthetic data generation, generating this data and evaluating it. The chosen algorithms have been:

- Gaussian Copula
- Generative Adversarial Network
- DataSynthesizer

Successfully it has been accomplished the first goal: generating synthetic data using different algorithms. The three of them have output three data sets, each of them, all with synthetic observations. The maximum number of synthesized data has been 150 observations, since it couldn't surpass the original dataset size (156 observations). Going over that threshold would mean snatching any kind of reality sense, given that over a 50% of data would be fake.

Nonetheless, this is not enough. It had to be proven that these algorithms are bringing back similar enough data. Checking this involved two more goals: having similar variables and having similar correlations.

With regard to evaluate how similar is the synthetic variable compared to the original one it's been designed a hypothesis test. A distribution is characterized by position and dispersion statistics, so it's been needed to check how close where the synthetic distribution statistics to the original ones. For each variable Levene's test has been performed, enabling variances' comparison. The test was a previous step for the u-test (Wilcoxon-Mann-Whitney test), if variances of both distributions were similar enough, then, median is tested. Every test's been done for a 95% confidence interval. In order to measure which performed better, the number of accepted hypothesis for each model and case has been counted, being the results shown in Table 5:

Synthetic Data	GC	GAN	DS
50	58,33%	37,50%	83,33%
100	70,83%	12,50%	95,83%
150	54,17%	16,67%	91,67%

Table 5: Percentage of accepted hypothesis for each model and number of synthesized data.

As it can be observed the model with higher percentage of accepted hypothesis is DS (83.33%-95.83%). GAN is the model with lowest acceptance ratio (16.67%-37.50%). While DS could keep very correctly original distributions and pass almost all hypothesis test, GAN did not, meaning that synthetic distributions did not look as alike as the confidence interval demanded.

The second aim of the evaluation, and third goal, was to check how the algorithm kept inter-variables relationships. A correlation matrix for each dataset has been computed, then original and synthetic data went under subtraction, ending up with a difference of correlations matrix. These matrices have gone under two processes for their analysis. Firstly, heatmaps for each case and algorithm were plotted, ideally these should be all white since our ideal correlations coefficients differences matrix should be made out of zeroes. Setting the scale from zero to the maximum difference, allows comparison between them because the color scale is the same for each plot, it's useful to understand visually the concept of the difference of correlation coefficients. Nonetheless it was needed a way to compare matrices in a quantifiable measure. That being the case, it's been computed the RSME for all of the correlation coefficient difference matrices. Even though this is usually a method for regression problems, it's been extrapolated to synthetic data generation, since it allows comparing which matrices were closer to zero. The result of the similarity analysis lead to conclude that, again, the best performer's been, overall, DS algorithm and the worst performer is, again, GAN model. GC is slightly better for 100 synthesized data, but this may be attributed to random error, since the algorithm has been just run once.

Now, results have been explained and summarized. It's clear to see that performance order for the chosen algorithms is:

1. DataSynthesizer
2. GaussianCopula
3. Generative Adversarial Network

Let's try to understand why models' order turned out like that. GAN models usually work very well when it comes to image generation, actually they do generate very realistic images, so why wasn't the best performer for tabular data? This is probably due to two reasons:

1. Data structure: images are much more structured when it comes to data, pixels are much more correlated between them than a table that's been created by a human, with variables that may be not correlated at all. This could be a difficulty for the GAN to learn the underlying patterns, since they usually are built with CNN (convolutional neural networks) and these models are well-suited learning the spatial relationships between pixels in an image.

2. Data abundance: is much easier to obtain a lot of images than, in this case, obtain a lot of valid data for Covid patients.

GaussianCopula model ended up in a second position, which is not bad, but it's important to understand why it's performance has not been the best. As it was explained earlier, Gaussian Copula model is a joint of Gaussian distributions, variables given for this exercise did not follow this distribution, so, in a way, they've been forced into the algorithm. This issue may lead to the algorithm not capturing more complex relationships between variables. Despite of this fact, the model's been able to preserve the correlations between variables and did a fair job while reproducing the variable's distributions themselves. SDV package allows the user to specify which type of distributions are each variable, unfortunately, this is out of the scope of this project and it haven't been looked into detail.

And last, but not least, it could have been expected that data synthesizer worked out the best. As a first reason, algorithm and code implementation already is much more complex, it needs more inputs and it's divided into two steps, a first data description and then a data generation based on that description. It also uses Bayesian statistics, which are based on the probabilities of different events happening. Developing a network based on that, may lead to think that it's not assuming any kind of distributions neither the data structure, it's just calculating chances of different events happening, giving a much more complex structure and fitted to the dataset.

The scope of this project could've been much more broader, but there's always time and resources limitations, that basically set the framework. In the next chapter, several approaches are proposed, amongst them, the original aim of this thesis.

8 Next Steps

This thesis is part of a long road where little steps made by students may make a big difference some day. Here are written down some proposal for next steps. Amongst them is written down what was the initial idea for this dissertation and why it's been changed. Here are a few extra proposals but, of course, there are infinite options as next steps to take.

8.1 Predictive model for different cases.

In the very beginning of this project it was aimed to bring the physicians a tool for ameliorating decision making. The main approach consisted in developing an algorithm that basically tells the professional, given a patient, if it's not intubated, returns survival rate (p_1) and if it goes over a set threshold, the same patient goes through a prediction model for they intubated and their survival rate (p_2). In the following paragraphs this procedure is detailed with graphics and further information.

8.1.1 Part I

This part consists on generating the datasets that will allow us develop both of the survival prediction models. This is explained using a flux diagram.

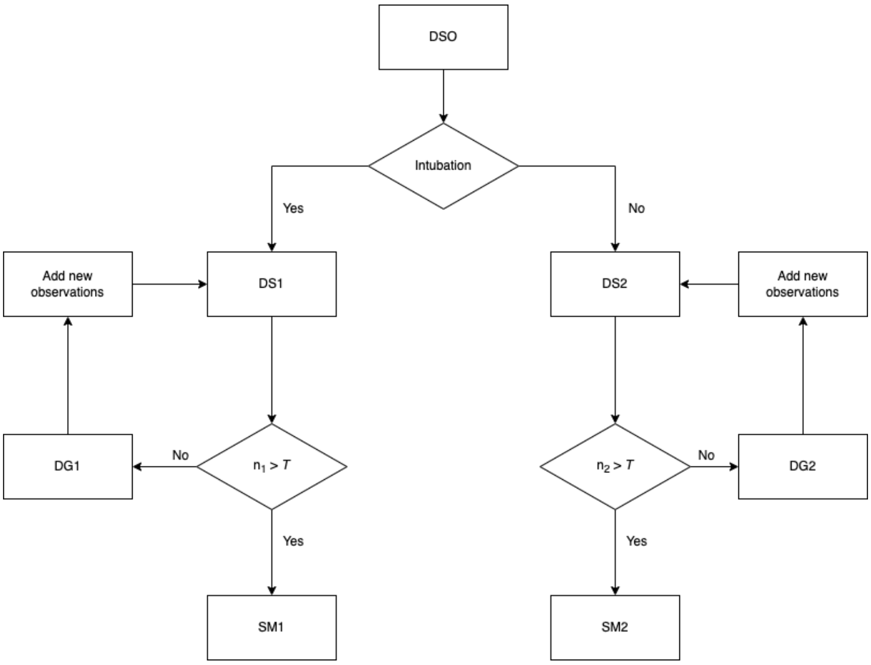


Figure 15: Flow diagram for the first part of the algorithm

The dataset needs to be split between those patients who've been intubated or have not been. Then for each subdataset (DS_1, DS_2), it's necessary to check that the number of rows (n_1, n_2) is higher than set threshold (T), meaning that if the number of rows is lower than the minimum we'd need to run a data generation model in order to expand this subdataset (DG_1, DG_2). Once T is surpassed, then the survival model for each class can be trained (SM_1, SM_2).

8.1.2 Part II

Now two survival models have been trained and each of them return the probability of dying of an *input* patient. This second part of the algorithm will be explained using a flow diagram too:

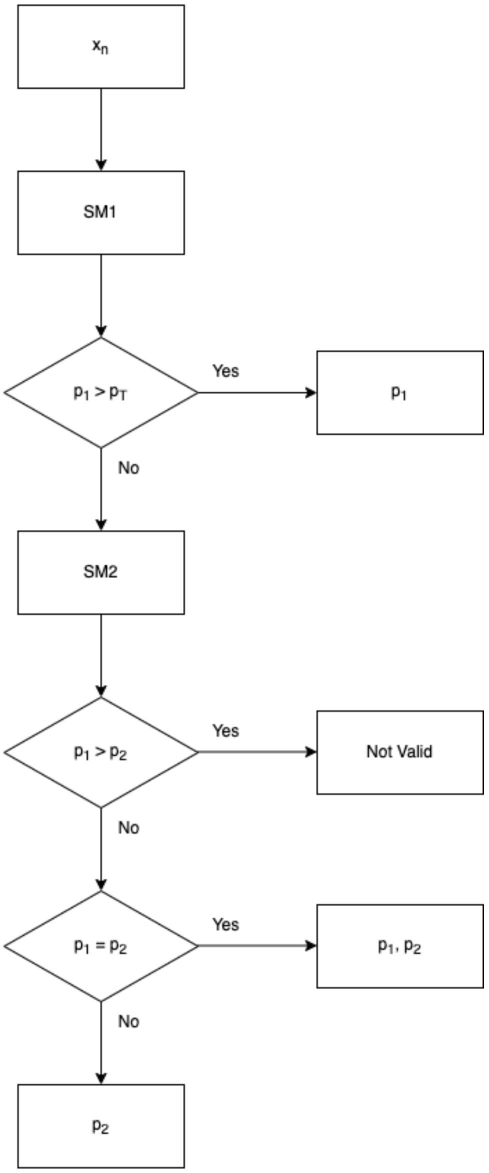


Figure 16: Flow diagram for the second part of the algorithm. Font: Original

So, let's suppose we receive a new input from a patient x_n, being the number of observations (rows) of the dataset (since it will be the last patient which the dataset will store the data from). Their data goes straight into the survival model for non-intubated patients (SM_1), if the probability (p_1) is greater than the set threshold probability (p_T), then the algorithm returns p_1, otherwise, the new patients data (x_n) steps into the survival model for intubated patients (SM_2), returning a new survival probability for the supposed intubated patient (p_2). If the intubation survival chances are less than non-intubation ones, then the output will be classified as *Not valid*, since it wouldn't make sense. If the two probabilities are equal, the algorithm should return them both as a result, since there's no space for an algorithm making a decision which one would be better. Then, if the chances of surviving being intubated are greater than non intubated chances, the algorithm returns p_2.

8.1.3 Limitations

This goal was highly ambitious, but it lead us to the very fundamentals: what should we do if there's no data? The main issue here was the lack of data. As it's been seen previously, the dataset was compound of 156 observations, and it was expected to train a model with less than 80 observations. The error propagation would have been enormous and the accuracy would have dropped abruptly. Choosing that path would have been extremly complicated and probably would have lead to no where, instead, it's been started from the beginning trying to solve this kind of limitations.

8.2 Improve Synthetic Data Models

The chosen models have a lot of parameters to explore. This was not on this thesis framework, but it would be interesting how much better can get these models if they're properly adjusted. Also, these are not the only models existing, a wider research can be done and other models can be compared to these ones.

8.3 Generating and testing data on bigger datasets

In this work the models have been able to recognise a pattern for 156 observations only, probably, as it has been mentioned earlier, this results wouldn't be totally reliable. Trying to use this models in bigger sets of data, end up having more observations, training models and compare differences between a model trained with real data and synthetic data, would be a really interesting project studying the scope of this field.

9 Economical and Environmental Analysis

This section presents the economic analysis, where the costs of the project are exposed, and the environmental analysis where it's analyzed how impacts the environment doing this project

9.1 Economical Analysis

Employee	€/h	h	€
Junior Engineer	14,08 €	600	8.448,00 €
Senior Engineer I	39,43 €	20	788,60 €
Senior Engineer II	39,43 €	20	788,60 €
Total			10.025,20 €

Table 6: Human resources cost

Supplies	Units	€/unit	€
Laptop	3	1.500,00 €	4.500,00 €
Total			4.500,00 €

Table 7: Material supplies cost

Supplies II	€	h	MW	€
Electricity	104,01 €/MWh	640	0,000061	4,06 €
Internet	0,07361 €/h	640		47,11 €
Total				51,17 €

Table 8: Electricity supplies cost

Total Budget	14.576,37 €

Table 9: Budget for the whole project

The costs have been split into: HHRR costs, how much money it would needed to be payed to the engineers working on the project; material supplies cost, how much can cost the working materials (in this case there's no more needed than a laptop); electricity costs, how much electricity[12][30] and internet[27] is used and how impacts economically the project.

9.2 Environmental Analysis

When it comes to the environmental analysis, it's been looked into the carbon footprint of the project[18][13]. The emissions for the usage of internet and laptop have been calculated and added up.

Environmental Impact	kgCO2/year	kgCO2/h	h	kgCO2
Laptop	88	0,01	640	6,43
Internet	474	0,05	640	34,63
Total				41,06

Table 10: Project's carbon footprint

An advantage of working in data projects is that the environmental footprint is very small, although, here are some recommendations [14] that may reduce the impact:

- Use your smartphone or other IT devices longer before upgrading, make sure to turn them off when it's not in use
- Use cloud-based services instead of storing data on physical devices to reduce the energy needed for data storage.
- Use digital platforms that utilizes renewable energy and investing in green energy.
- Avoid unnecessary printing and use digital means of communication and documentation.

9.3 Social Impact

As it's been mentioned in previous chapters, the origin of the data lack tackle is in the healthcare system, therefore, this dissertation could have also a social impact. Here are some points about how data generation can help data scientists in this field:

- Privatization of data. Working public data is always a bureaucratic and ethical nightmare, being able to generate very similar data to the original helps both sides of the equation: scientists who need those values and patients' privacy, since the real values won't be leaked.
- Deals with data lack barrier. Having the chance to have more observations without actually having them is removing a very big obstacle for a lot of scientists, let it be for educational or academical purposes or for real applications. Developing more accurate tools could be possible thanks to this, which would mean, in a future, a better management of clinical resources.

References

[1] Miguel Arquez Abdala. *Métodos de resampling*. [Accessed 03-Jan-2023]. Mar. 2020. URL: https://rstudio-pubs-static.s3.amazonaws.com/591883_d3879f873d3a4594bffe25b html.

[2] Joaquin Amat. *Test de Wilcoxon Mann Whitney como alternativa al t-test*. URL: https://www.cienciadedatos.net/documentos/17_mann%E2%80%93whitney_u_test. (accessed: 29.12.2022).

[3] Joaquin Amat. *Test U de Mann-Whitney-Wilcoxon (U-test) con Python*. URL: https://www.cienciadedatos.net/documentos/pystats10-t-test-python.html. (accessed: 29.12.2022).

[4] Fodil Benali et al. "MTCopula: Synthetic Complex Data Generation Using Copula". In: *23rd International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP)*. Nicosia, Cyprus, 2021, pp. 51–60. URL: https://hal.archives-ouvertes.fr/hal-03188317.

[5] J. Brownlee. *How Much Training Data is Required for Machine Learning?* [Accessed 10-1-2023]. URL: https://machinelearningmastery.com/much-training-data-required-machine-learning/.

[6] *CDC Museum COVID-19 Timeline*. URL: https://www.cdc.gov/museum/timeline/covid19.html. (accessed: 12.11.2022).

[7] N. V. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: (2011). DOI: 10.48550/ARXIV.1106.1813. URL: https://arxiv.org/abs/1106.1813.

[8] Weron, R. Cízek, P. Härdle, W. *17.1 Copulas*. [Accessed 19-12-2022]. Mar. 2005. URL: http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/tutorials/sfehtmlnode86.html.

[9] Collaborative. *Copula (probability theory)*. URL: https://en.wikipedia.org/wiki/Copula_(probability_theory). (accessed: 19.12.2022).

[10] DataCebo. *The Synthetic Data Vault Project*. Version latest. Dec. 2022. DOI: 10.1109/DSAA.2016.49. URL: https://sdv.dev/SDV/.

[11] Google Developers. *Background: What is a Generative Model?* [Accessed 19-12-2022]. July 2022. URL: https://developers.google.com/machine-learning/gan/generative?hl=en.

[12] Trading Economics. *Spain Electricity Price - 2023 Data - 1998-2022 Historical - 2024 Forecast*. https://tradingeconomics.com/spain/electricity-price. [Accessed 03-Jan-2023]. 2023.

[13] EnerGuide. *How much power does a computer use? And how much CO2 does that represent?* https://www.energuide.be/en/questions-answers/how-much-power-does-a-computer-use-and-how-much-co2-does-that-represent/54/. [Accessed 03-Jan-2023].

[14] Ericsson. *A quick guide to your digital carbon footprint carbon footprint*. https://www.ericsson.com/4ac671/assets/local/reports-papers/consumerlab/reports/2020/ericsson-true-or-false-report-screen.pdf. [Accessed 03-Jan-2023]. 2020.

[15] Sheri Fink. “Worst-Case Estimates for U.S. Coronavirus Deaths”. In: *The New York Times* (Mar. 13, 2020). URL: https://www.nytimes.com/2020/03/13/us/coronavirus-deaths-estimate.html (visited on 11/12/2022).

[16] S.E. Galaitsi et al. “The challenges of data usage for the United States’ COVID-19 response”. In: *International Journal of Information Management* 59 (Aug. 2021), p. 102352. DOI: 10.1016/j.ijinfomgt.2021.102352. URL: https://doi.org/10.1016/j.ijinfomgt.2021.102352.

[17] TRACIE. Healthcare Emergency Preparedness Information Getaway, ed. *SOFA Score: What it is and How to Use it in Triage*. Dec. 21, 2020. URL: https://files.asprtracie.hhs.gov/documents/aspr-tracie-sofa-score-fact-sheet.pdf.

[18] Sarah Griffiths. *Why your internet habits are not as clean as you think*. https://www.bbc.com/future/article/20200305-why-your-internet-habits-are-not-as-clean-as-you-think. [Accessed 08-Jan-2023]. 2020.

[19] Michal Horný. *Bayesian Networks*. Tech. rep. Boston University School of Public Health, 2014.

[20] Amanda Kobokovich. *Ventilator Stockpiling and Availability in the US*. Mar. 9, 2020. URL: https://www.centerforhealthsecurity.org/resources/COVID-19/COVID-19-fact-sheets/200214-VentilatorAvailability-factsheet.pdf.

[21] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[22] David Meyer, Thomas Nagler, and Robin J. Hogan. “Copula-based synthetic data generation for machine learning emulators in weather and climate: application to a simple radiation model”. In: *CoRR* abs/2012.09037 (2020). arXiv: 2012.09037. URL: https://arxiv.org/abs/2012.09037.

[23] Arslaan Hamza Mohammad Jabeen. “Analysis and data processing to identify Covid-19 patients with spontaneous respiratory capacity and predict their mortality”. PhD thesis. Universitat Politècnica de Catalunya, 2022.

[24] Muthukrishnan. *Understanding Correlations and Correlation Matrix*. https://muthu.co/understanding-correlations-and-correlation-matrix/. [Accessed 01-Jan-2023]. 2021.

[25] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. “The Synthetic Data Vault”. In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 2016, pp. 399–410. DOI: 10.1109/DSAA.2016.49.

[26] Haoyue Ping, Julia Stoyanovich, and Bill Howe. “DataSynthesizer”. In: *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*. Chicago IL USA: ACM, June 2017.

[27] Portaltic. *Los españoles pagan de media un 27,4% más por Internet que el resto de europeos*. https://www.europapress.es/portaltic/internet/noticia-espanoles-pagan-media-274-mas-internet-resto-europeos-20181020112938.html. [Accessed 08-Jan-2023]. 2018.

[28] Junaid Rehman. *Advantages and disadvantages of generative adversarial networks (GAN)*. [Accessed 19-12-2022]. URL: https://www.itrelease.com/2020/06/advantages-and-disadvantages-of-generative-adversarial-networks-gan/.

[29] Robert D. Truog, M.D., Christine Mitchell, R.N., and George Q. Daley, M.D., Ph.D. “The Toughest Triage — Allocating Ventilators in a Pandemic.” In: *The New England Journal of Medicine* 382.21 (2020), pp. 1973–1975. DOI: https://doi.org/10.1056/NEJMp2005689.

[30] Eco Cost Savings. *How Many Watts Does A Laptop Use? [Actual Usage & Costs Revealed*. https://ecocostsavings.com/how-many-watts-does-a-laptop-use/. [Accessed 03-Jan-2023]. 2021.

[31] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[32] Unknown. *GAN*. URL: https://developers.google.com/machine-learning/gan?hl=en. (accessed: 22.12.2022).

[33] Basil Varkey. "Principles of Clinical Ethics and Their Application to Practice." In: *Karger* 30.1 (2020), pp. 17–28. DOI: https://doi.org/10.1159/000509119.

[34] Wikipedia. *Bayesian network — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Bayesian%20network&oldid=1094972270. [Online; accessed 01-January-2023]. 2023.

[35] Wikipedia. *Levene's test — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Levene's%20test&oldid=1120031847. [Online; accessed 01-January-2023]. 2023.

[36] Wikipedia. *Root-mean-square deviation — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Root-mean-square%20deviation&oldid=1122929780. [Online; accessed 13-January-2023]. 2023.

[37] Lei Xu et al. "Modeling Tabular data using Conditional GAN". In: *CoRR* abs/1907.00503 (2019). arXiv: 1907.00503. URL: http://arxiv.org/abs/1907.00503.

APPENDIX

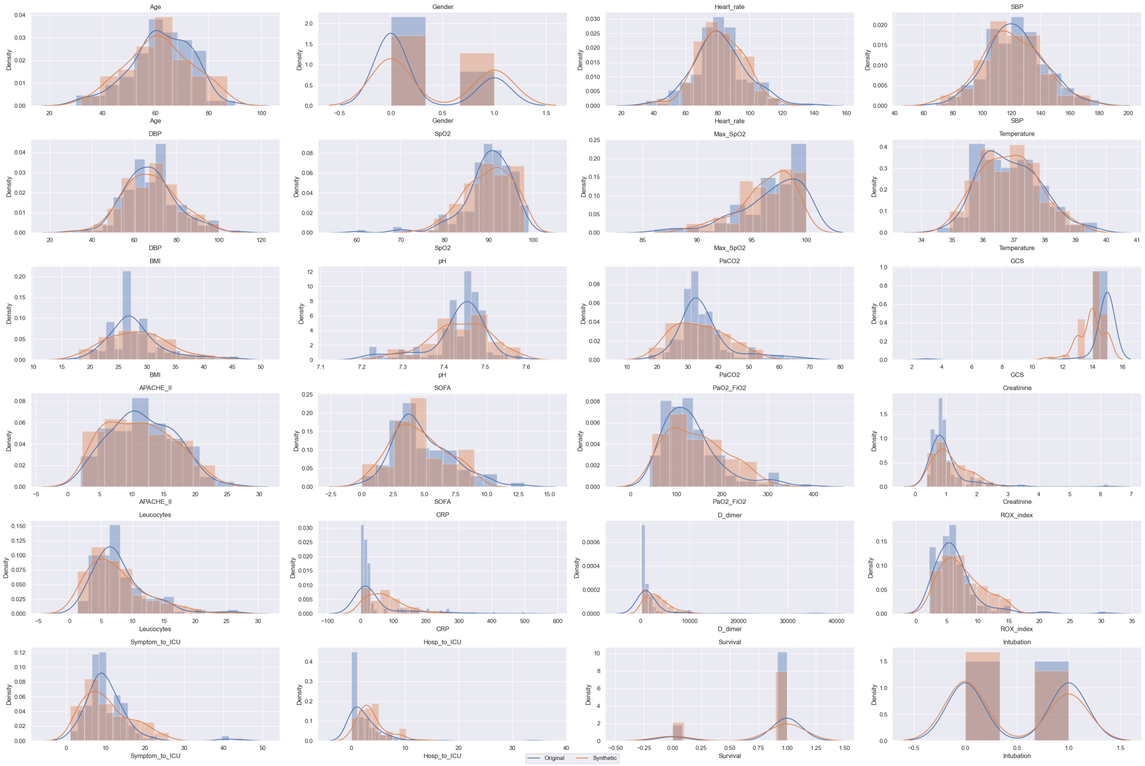
Code

Scripts and code aren't available due to confidentiality reasons.

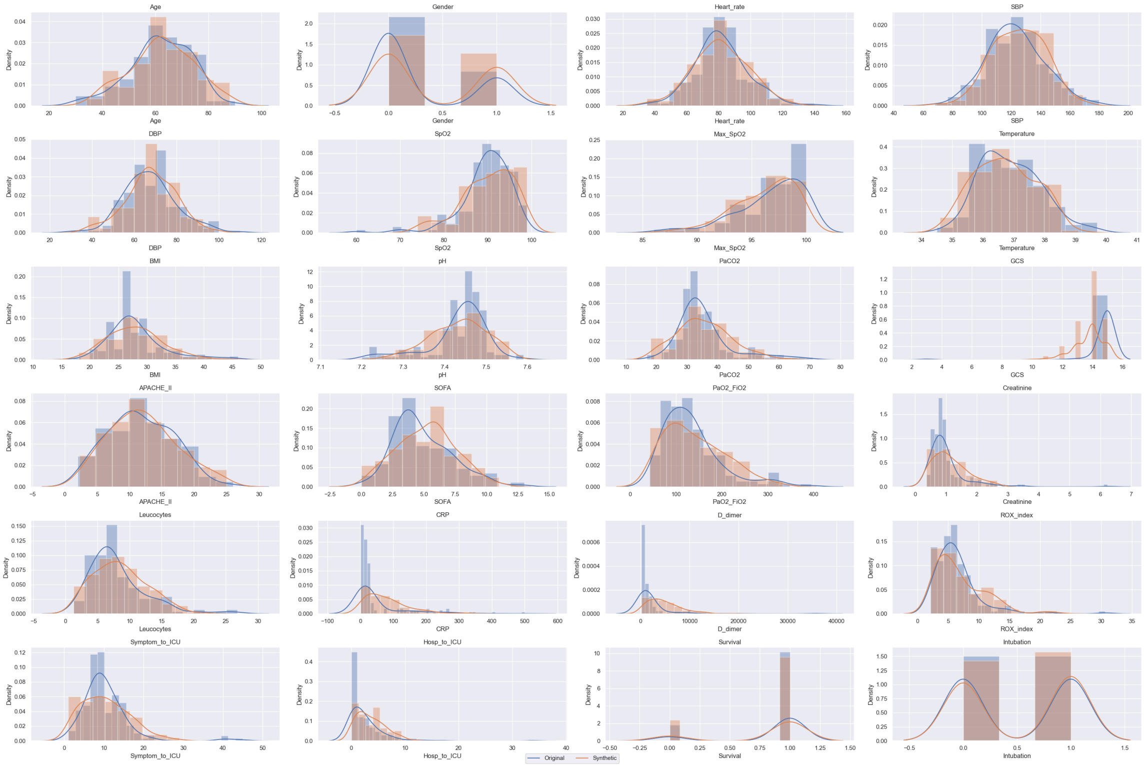
Plots

Here can be found distribution plots for each algorithm and 100 and 150 synthesized observations cases.

Gaussian Copula

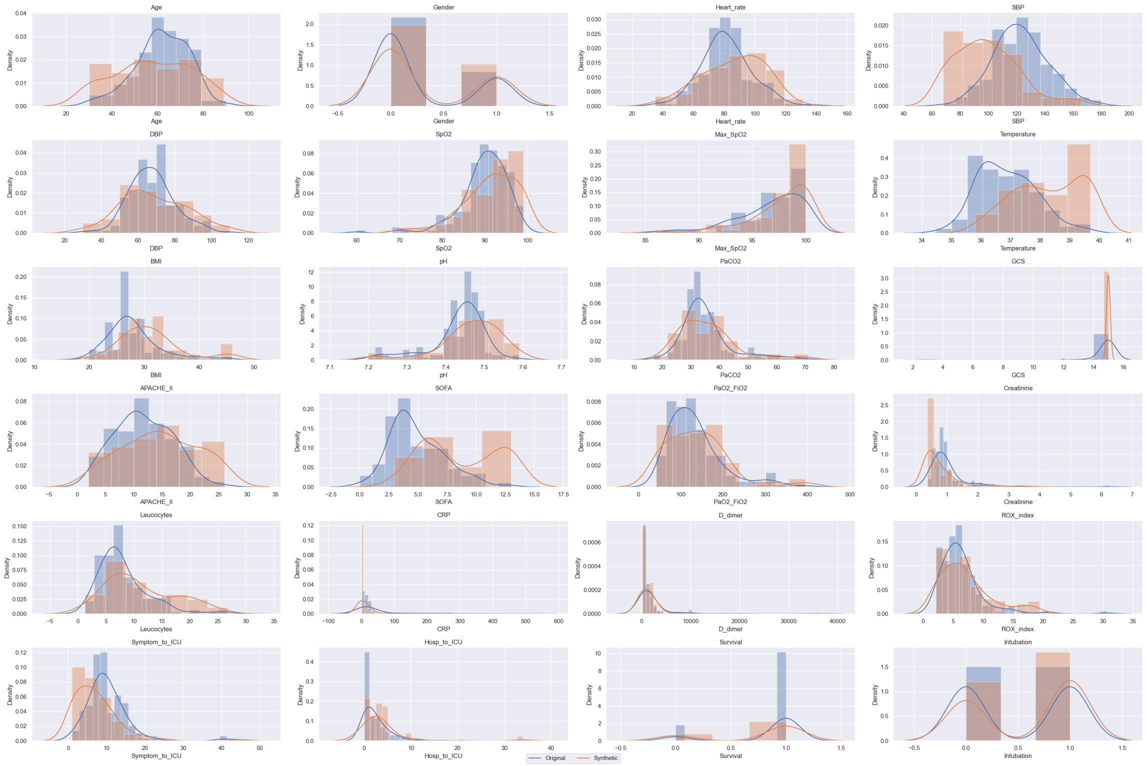


Plot distribution comparison for 100 synthetic data

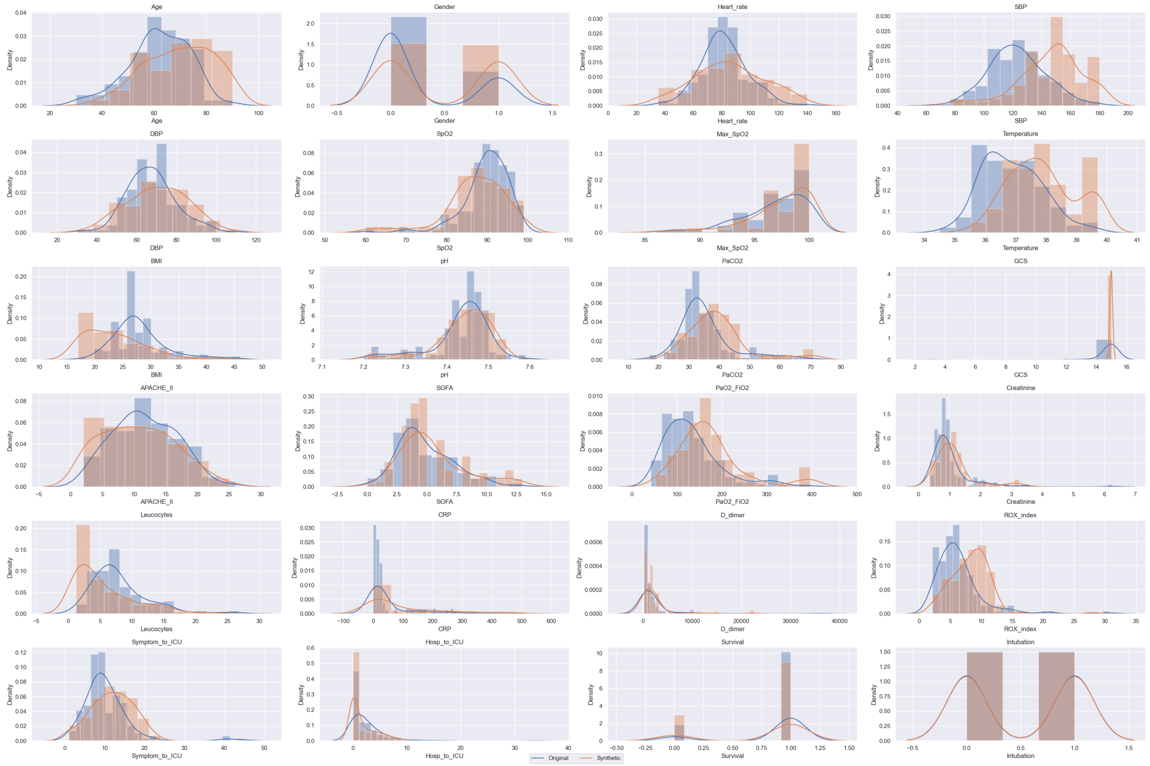


Plot distribution comparison for 150 synthetic data

Generative Adversarial Network

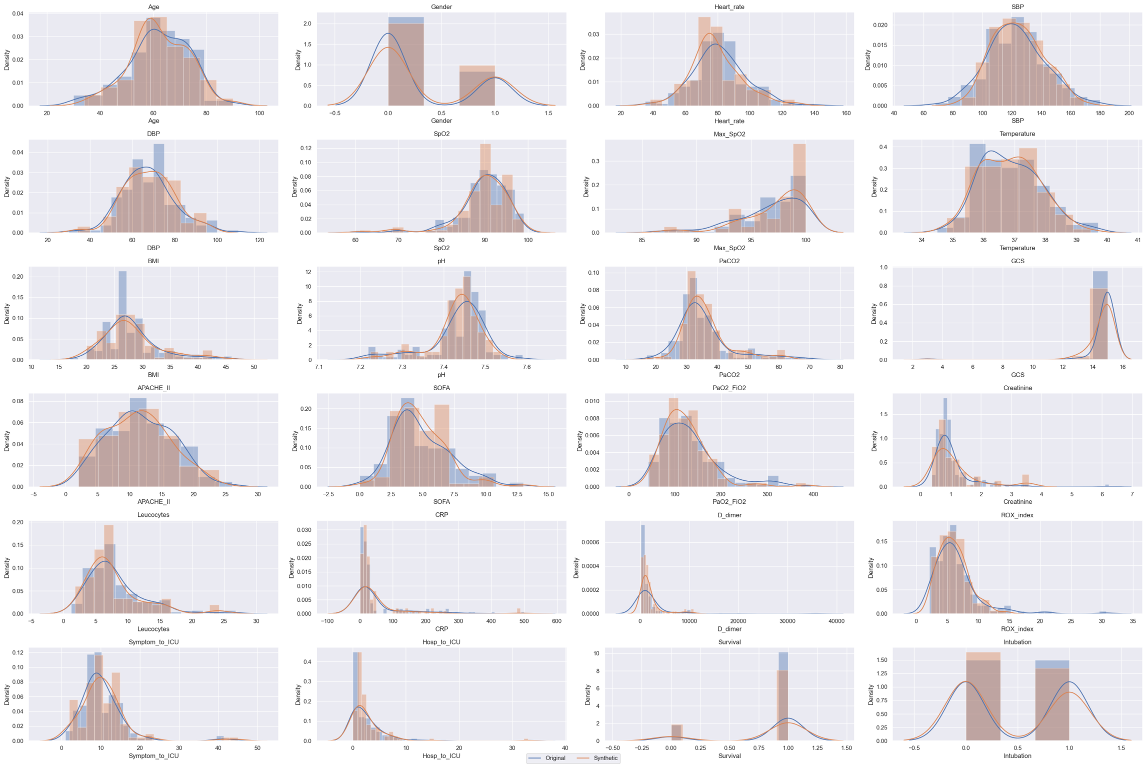


Plot distribution comparison for 100 synthetic data

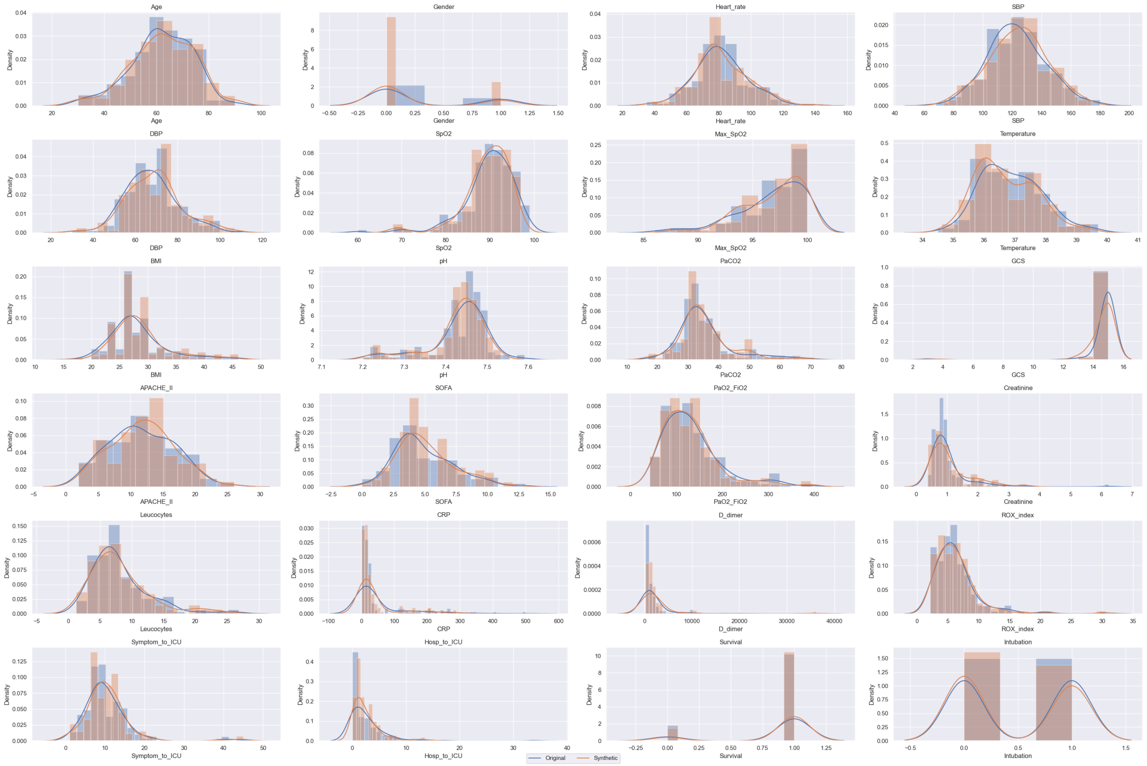


Plot distribution comparison for 150 synthetic data

DataSynthesizer



Plot distribution comparison for 100 synthetic data



Plot distribution comparison for 150 synthetic data

Tables

Hypothesis Test

Gaussian Copula

Variable	W	pval	equal_var	U-val	p-val	RBC	CLES	result
Age	2,55	0,11	TRUE	3.677,00	0,46	0,07	0,47	Accept
Gender	1,20	0,27	TRUE	3.628,00	0,27	0,08	0,46	Accept
Heart_rate	0,60	0,44	TRUE	4.336,00	0,30	-0,10	0,55	Accept
SBP	2,08	0,15	TRUE	4.134,50	0,62	-0,05	0,52	Accept
DBP	0,65	0,42	TRUE	4.899,00	0,01	-0,24	0,62	Reject
SpO2	0,87	0,35	TRUE	4.796,00	0,02	-0,21	0,61	Reject
Max_SpO2	0,22	0,64	TRUE	4.605,50	0,08	-0,17	0,58	Accept
Temperature	0,26	0,61	TRUE	3.962,00	0,98	0,00	0,50	Accept
BMI	0,81	0,37	TRUE	3.498,00	0,22	0,11	0,44	Accept
pH	5,01	0,03	FALSE	4.156,00	0,58	-0,05	0,53	Reject
PaCO2	0,97	0,33	TRUE	4.489,50	0,15	-0,14	0,57	Accept
GCS	2,10	0,15	TRUE	6.361,50	0,00	-0,61	0,81	Reject
APACHE_II	0,74	0,39	TRUE	3.926,50	0,95	0,01	0,50	Accept
SOFA	0,08	0,78	TRUE	3.624,50	0,38	0,08	0,46	Accept
PaO2_FiO2	1,54	0,22	TRUE	3.862,00	0,81	0,02	0,49	Accept

Creatinine	1,11	0,29	TRUE	2.936,00	0,01	0,26	0,37	Reject
Leucocytes	1,45	0,23	TRUE	4.712,00	0,04	-0,19	0,60	Reject
CRP	0,01	0,93	TRUE	2.131,00	0,00	0,46	0,27	Reject
D_dimer	2,04	0,16	TRUE	1.610,50	0,00	0,59	0,20	Reject
ROX_index	2,09	0,15	TRUE	3.815,00	0,72	0,03	0,48	Accept
Symptom_to_ICU	3,18	0,08	TRUE	4.140,00	0,61	-0,05	0,52	Accept
Hosp_to_ICU	0,21	0,65	TRUE	2.839,50	0,00	0,28	0,36	Reject
Survival	5,56	0,02	FALSE	4.535,00	0,02	-0,15	0,57	Reject
Intubation	0,25	0,62	TRUE	4.029,00	0,81	-0,02	0,51	Accept

Gaussian Copula hypothesis test results for 50 new synthetic data.

Variable	**W**	**pval**	**equal_var**	**U-val**	**p-val**	**RBC**	**CLES**	**results**
Age	0,17	0,68	TRUE	8.018,00	0,84	-0,02	0,51	Accept
Gender	10,15	0,00	FALSE	6.387,00	0,00	0,19	0,40	Reject
Heart_rate	0,49	0,49	TRUE	7.600,00	0,61	0,04	0,48	Accept
SBP	0,31	0,58	TRUE	7.254,50	0,27	0,08	0,46	Accept
DBP	0,09	0,77	TRUE	7.445,50	0,44	0,06	0,47	Accept
SpO2	2,22	0,14	TRUE	8.311,50	0,48	-0,05	0,53	Accept
Max_SpO2	0,24	0,62	TRUE	8.926,00	0,08	-0,13	0,57	Accept
Temperature	0,79	0,38	TRUE	8.221,00	0,58	-0,04	0,52	Accept
BMI	1,79	0,18	TRUE	6.871,00	0,08	0,13	0,44	Accept
pH	0,00	1,00	TRUE	8.234,50	0,57	-0,04	0,52	Accept
PaCO2	0,88	0,35	TRUE	6.947,50	0,10	0,12	0,44	Accept
GCS	5,12	0,02	FALSE	13.137,00	0,00	-0,66	0,83	Reject
APACHE_II	1,56	0,21	TRUE	7.839,50	0,92	0,01	0,50	Accept
SOFA	0,04	0,83	TRUE	7.313,00	0,31	0,07	0,46	Accept
PaO2_FiO2	0,33	0,56	TRUE	8.060,50	0,78	-0,02	0,51	Accept
Creatinine	1,96	0,16	TRUE	6.233,50	0,00	0,21	0,40	Reject
Leucocytes	1,57	0,21	TRUE	8.024,00	0,83	-0,02	0,51	Accept
CRP	0,72	0,40	TRUE	4.428,00	0,00	0,44	0,28	Reject
D_dimer	0,69	0,41	TRUE	3.059,00	0,00	0,61	0,19	Reject
ROX_index	0,22	0,64	TRUE	7.572,00	0,58	0,04	0,48	Accept
Symptom_to_ICU	8,63	0,00	FALSE	7.332,00	0,33	0,07	0,46	Reject
Hosp_to_ICU	4,07	0,05	FALSE	4.854,00	0,00	0,39	0,31	Reject
Survival	1,43	0,23	TRUE	8.359,00	0,23	-0,06	0,53	Accept
Intubation	3,13	0,08	TRUE	8.453,00	0,27	-0,07	0,54	Accept

Gaussian Copula hypothesis test results for 100 new synthetic data.

Variable	**W**	**pval**	**equal_var**	**U-val**	**p-val**	**RBC**	**CLES**	**results**
Age	0,81	0,37	TRUE	12.361,50	0,51	-0,04	0,52	Accept
Gender	6,29	0,01	FALSE	10.252,00	0,01	0,14	0,43	Reject

Heart_rate	1,62	0,20	TRUE	10.986,00	0,27	0,07	0,46	Accept
SBP	0,78	0,38	TRUE	11.765,50	0,91	0,01	0,50	Accept
DBP	0,28	0,60	TRUE	12.093,50	0,76	-0,02	0,51	Accept
SpO2	6,38	0,01	FALSE	12.338,00	0,53	-0,04	0,52	Reject
Max_SpO2	0,74	0,39	TRUE	13.121,50	0,10	-0,11	0,55	Accept
Temperature	0,08	0,78	TRUE	10.071,50	0,02	0,15	0,43	Reject
BMI	0,61	0,44	TRUE	12.604,00	0,33	-0,06	0,53	Accept
pH	6,50	0,01	FALSE	12.603,50	0,34	-0,06	0,53	Reject
PaCO2	6,37	0,01	FALSE	11.071,00	0,32	0,07	0,47	Reject
GCS	5,42	0,02	FALSE	18.990,50	0,00	-0,60	0,80	Reject
APACHE_II	0,07	0,80	TRUE	12.528,50	0,39	-0,06	0,53	Accept
SOFA	0,01	0,91	TRUE	11.537,50	0,69	0,03	0,49	Accept
PaO2_FiO2	2,72	0,10	TRUE	12.043,00	0,81	-0,02	0,51	Accept
Creatinine	4,19	0,04	FALSE	11.087,00	0,33	0,06	0,47	Reject
Leucocytes	0,79	0,38	TRUE	11.736,00	0,88	0,01	0,50	Accept
CRP	0,00	0,96	TRUE	6.223,50	0,00	0,48	0,26	Reject
D_dimer	1,77	0,19	TRUE	5.605,00	0,00	0,53	0,24	Reject
ROX_index	2,00	0,16	TRUE	11.856,00	0,99	0,00	0,50	Accept
Symptom_to_ICU	7,67	0,01	FALSE	11.772,00	0,92	0,01	0,50	Reject
Hosp_to_ICU	0,10	0,75	TRUE	8.839,00	0,00	0,25	0,37	Reject
Survival	0,92	0,34	TRUE	12.341,00	0,34	-0,04	0,52	Accept
Intubation	0,11	0,74	TRUE	11.692,00	0,82	0,01	0,49	Accept

Gaussian Copula hypothesis test results for 150 new synthetic data.

Generative Adversarial Network

Variable	**W**	**pval**	**equal_var**	**U-val**	**p-val**	**RBC**	**CLES**	**result**
Age	5,39	0,02	FALSE	2.432,50	0,00	0,38	0,31	Reject
Gender	4,63	0,03	FALSE	3.312,00	0,03	0,16	0,42	Reject
Heart_rate	11,27	0,00	FALSE	3.760,00	0,61	0,05	0,48	Reject
SBP	11,95	0,00	FALSE	5.747,00	0,00	-0,46	0,73	Reject
DBP	35,86	0,00	FALSE	2.503,00	0,00	0,37	0,32	Reject
SpO2	0,14	0,71	TRUE	4.546,50	0,11	-0,15	0,58	Accept
Max_SpO2	0,23	0,63	TRUE	3.232,50	0,05	0,18	0,41	Accept
Temperature	1,03	0,31	TRUE	4.145,00	0,60	-0,05	0,53	Accept
BMI	5,88	0,02	FALSE	5.213,50	0,00	-0,32	0,66	Reject
pH	0,22	0,64	TRUE	2.716,50	0,00	0,31	0,34	Reject
PaCO2	6,26	0,01	FALSE	3.870,00	0,83	0,02	0,49	Reject
GCS	1,56	0,21	TRUE	3.650,00	0,05	0,08	0,46	Reject
APACHE_II	2,66	0,11	TRUE	4.594,00	0,08	-0,16	0,58	Accept
SOFA	2,22	0,14	TRUE	2.536,00	0,00	0,36	0,32	Reject
PaO2_FiO2	0,89	0,35	TRUE	4.729,00	0,04	-0,20	0,60	Reject
Creatinine	0,01	0,95	TRUE	3.886,00	0,86	0,02	0,49	Accept

Leucocytes	4,02	0,05	FALSE	3.905,50	0,91	0,01	0,49	Reject
CRP	3,84	0,05	TRUE	2.928,00	0,01	0,26	0,37	Reject
D_dimer	2,70	0,10	TRUE	3.343,50	0,10	0,15	0,42	Accept
ROX_index	1,38	0,24	TRUE	4.292,00	0,36	-0,09	0,54	Accept
Symptom_to_ICU	1,17	0,28	TRUE	3.281,00	0,07	0,17	0,42	Accept
Hosp_to_ICU	5,73	0,02	FALSE	4.728,50	0,03	-0,20	0,60	Reject
Survival	2,06	0,15	TRUE	4.298,00	0,15	-0,09	0,54	Accept
Intubation	6,52	0,01	FALSE	3.555,00	0,22	0,10	0,45	Reject

Generative Adversarial Network hypothesis test results for 50 new synthetic data.

Variable	W	pval	equal_var	U-val	p-val	RBC	CLES	result
Age	0,08	0,78	TRUE	2.541,50	0,00	0,68	0,16	Reject
Gender	4,85	0,03	FALSE	6.861,00	0,03	0,13	0,43	Reject
Heart_rate	14,70	0,00	FALSE	7.807,00	0,87	0,01	0,49	Reject
SBP	7,44	0,01	FALSE	8.436,50	0,36	-0,07	0,53	Reject
DBP	3,50	0,06	TRUE	11.902,00	0,00	-0,51	0,75	Reject
SpO2	9,53	0,00	FALSE	9.356,00	0,01	-0,18	0,59	Reject
Max_SpO2	7,08	0,01	FALSE	9.683,00	0,00	-0,23	0,61	Reject
Temperature	4,08	0,04	FALSE	9.063,00	0,05	-0,15	0,57	Reject
BMI	4,01	0,05	FALSE	9.702,50	0,00	-0,23	0,61	Reject
pH	0,79	0,38	TRUE	11.230,50	0,00	-0,42	0,71	Reject
PaCO2	0,86	0,36	TRUE	9.892,50	0,00	-0,25	0,63	Reject
GCS	2,61	0,11	TRUE	7.380,50	0,02	0,07	0,47	Reject
APACHE_II	3,71	0,06	TRUE	8.770,50	0,14	-0,11	0,56	Accept
SOFA	1,07	0,30	TRUE	12.096,50	0,00	-0,53	0,77	Reject
PaO2_FiO2	11,50	0,00	FALSE	6.912,00	0,09	0,13	0,44	Reject
Creatinine	1,75	0,19	TRUE	11.734,50	0,00	-0,49	0,74	Reject
Leucocytes	3,18	0,08	TRUE	12.658,50	0,00	-0,60	0,80	Reject
CRP	12,77	0,00	FALSE	12.996,00	0,00	-0,65	0,82	Reject
D_dimer	0,55	0,46	TRUE	6.939,00	0,10	0,12	0,44	Accept
ROX_index	6,38	0,01	FALSE	5.794,50	0,00	0,27	0,37	Reject
Symptom_to_ICU	1,29	0,26	TRUE	13.329,00	0,00	-0,69	0,84	Reject
Hosp_to_ICU	2,55	0,11	TRUE	6.264,50	0,01	0,21	0,40	Reject
Survival	7,30	0,01	FALSE	8.991,00	0,01	-0,14	0,57	Reject
Intubation	0,57	0,45	TRUE	7.663,00	0,64	0,03	0,49	Accept

Generative Adversarial Network hypothesis test results for 100 new synthetic data.

Variable	W	pval	equal_var	U-val	p-val	RBC	CLES	result
Age	4,45	0,04	FALSE	7.384,00	0,00	0,38	0,31	Reject
Gender	5,13	0,02	FALSE	10.410,00	0,02	0,12	0,44	Reject
Heart_rate	14,61	0,00	FALSE	17.836,00	0,00	-0,51	0,75	Reject

SBP	8,12	0,01	FALSE	8.577,50	0,00	0,28	0,36	Reject
DBP	6,62	0,01	FALSE	5.560,50	0,00	0,53	0,24	Reject
SpO2	10,99	0,00	FALSE	10.199,00	0,03	0,14	0,43	Reject
Max_SpO2	0,50	0,48	TRUE	12.899,00	0,18	-0,09	0,54	Accept
Temperature	0,05	0,82	TRUE	19.629,50	0,00	-0,66	0,83	Reject
BMI	1,55	0,21	TRUE	8.572,50	0,00	0,28	0,36	Reject
pH	1,89	0,17	TRUE	14.209,00	0,00	-0,20	0,60	Reject
PaCO2	2,67	0,10	TRUE	18.483,00	0,00	-0,56	0,78	Reject
GCS	4,70	0,03	FALSE	10.950,00	0,00	0,08	0,46	Reject
APACHE_II	2,73	0,10	TRUE	7.919,50	0,00	0,33	0,33	Reject
SOFA	3,82	0,05	TRUE	14.573,50	0,00	-0,23	0,62	Reject
PaO2_FiO2	0,68	0,41	TRUE	16.609,50	0,00	-0,40	0,70	Reject
Creatinine	6,68	0,01	FALSE	19.368,00	0,00	-0,63	0,82	Reject
Leucocytes	3,98	0,05	FALSE	18.474,00	0,00	-0,56	0,78	Reject
CRP	35,42	0,00	FALSE	5.702,50	0,00	0,52	0,24	Reject
D_dimer	0,28	0,60	TRUE	9.599,50	0,00	0,19	0,41	Reject
ROX_index	19,22	0,00	FALSE	10.861,50	0,21	0,08	0,46	Reject
Symptom_to_ICU	1,30	0,26	TRUE	10.942,50	0,24	0,08	0,46	Accept
Hosp_to_ICU	0,58	0,45	TRUE	12.546,00	0,36	-0,06	0,53	Accept
Survival	5,59	0,02	FALSE	13.131,00	0,02	-0,11	0,55	Reject
Intubation	1,01	0,32	TRUE	11.376,00	0,48	0,04	0,48	Accept

Generative Adversarial Network hypothesis test results for 150 new synthetic data.

DataSynthesizer

Variable	W	pval	equal_var	U-val	p-val	RBC	CLES	result
Age	0,24	0,62	TRUE	4.157,00	0,58	-0,05	0,53	Accept
Gender	0,66	0,42	TRUE	4.181,00	0,42	-0,06	0,53	Accept
Heart_rate	1,00	0,32	TRUE	2.974,50	0,01	0,25	0,38	Reject
SBP	0,11	0,75	TRUE	3.588,50	0,33	0,09	0,45	Accept
DBP	0,36	0,55	TRUE	4.061,50	0,76	-0,03	0,51	Accept
SpO2	0,13	0,72	TRUE	4.083,50	0,72	-0,03	0,52	Accept
Max_SpO2	0,10	0,76	TRUE	4.452,00	0,17	-0,13	0,56	Accept
Temperature	0,33	0,57	TRUE	3.969,00	0,96	-0,01	0,50	Accept
BMI	0,12	0,73	TRUE	3.843,50	0,77	0,03	0,49	Accept
pH	1,43	0,23	TRUE	4.492,00	0,14	-0,14	0,57	Accept
PaCO2	0,04	0,84	TRUE	3.564,00	0,30	0,10	0,45	Accept
GCS	1,11	0,29	TRUE	4.650,50	0,00	-0,18	0,59	Reject
APACHE_II	0,76	0,39	TRUE	3.630,00	0,39	0,08	0,46	Accept
SOFA	0,00	0,95	TRUE	3.996,50	0,90	-0,01	0,51	Accept
PaO2_FiO2	0,14	0,71	TRUE	4.229,50	0,45	-0,07	0,54	Accept
Creatinine	0,15	0,70	TRUE	4.324,00	0,31	-0,10	0,55	Accept
Leucocytes	0,25	0,62	TRUE	3.856,00	0,80	0,02	0,49	Accept

CRP	1,61	0,21	TRUE	3.124,00	0,03	0,21	0,40	Reject
D_dimer	0,05	0,83	TRUE	3.596,50	0,34	0,09	0,46	Accept
ROX_index	0,39	0,53	TRUE	4.602,00	0,08	-0,17	0,58	Accept
Symptom_to_ICU	0,01	0,94	TRUE	3.861,50	0,81	0,02	0,49	Accept
Hosp_to_ICU	0,04	0,84	TRUE	4.059,00	0,77	-0,03	0,51	Accept
Survival	0,85	0,36	TRUE	3.745,00	0,36	0,05	0,47	Accept
Intubation			FALSE	3.950,00	1,00	0,00	0,50	Reject

DataSynthesizer hypothesis test results for 50 new synthetic data.

Variable	**W**	**pval**	**equal_var**	**U-val**	**p-val**	**RBC**	**CLES**	**result**
Age	0,56	0,46	TRUE	7.996,50	0,87	-0,01	0,51	Accept
Gender	0,77	0,38	TRUE	7.493,00	0,38	0,05	0,47	Accept
Heart_rate	0,79	0,38	TRUE	8.697,00	0,17	-0,10	0,55	Accept
SBP	1,10	0,30	TRUE	7.536,50	0,53	0,05	0,48	Accept
DBP	0,04	0,84	TRUE	7.517,50	0,51	0,05	0,48	Accept
SpO2	0,00	0,97	TRUE	7.917,50	0,98	0,00	0,50	Accept
Max_SpO2	1,25	0,26	TRUE	7.142,00	0,19	0,10	0,45	Accept
Temperature	0,09	0,77	TRUE	8.021,00	0,84	-0,02	0,51	Accept
BMI	0,39	0,53	TRUE	8.059,00	0,79	-0,02	0,51	Accept
pH	1,30	0,26	TRUE	8.647,00	0,20	-0,10	0,55	Accept
PaCO2	0,84	0,36	TRUE	7.052,00	0,15	0,11	0,45	Accept
GCS	0,98	0,32	TRUE	9.161,00	0,00	-0,16	0,58	Reject
APACHE_II	0,01	0,94	TRUE	8.298,00	0,50	-0,05	0,53	Accept
SOFA	1,45	0,23	TRUE	7.980,50	0,89	-0,01	0,51	Accept
PaO2_FiO2	3,41	0,07	TRUE	8.469,50	0,33	-0,07	0,54	Accept
Creatinine	2,76	0,10	TRUE	7.926,00	0,97	0,00	0,50	Accept
Leucocytes	0,12	0,73	TRUE	8.366,00	0,43	-0,06	0,53	Accept
CRP	0,51	0,48	TRUE	7.516,00	0,51	0,05	0,48	Accept
D_dimer	0,94	0,33	TRUE	6.840,00	0,07	0,13	0,43	Accept
ROX_index	1,97	0,16	TRUE	7.733,00	0,78	0,02	0,49	Accept
Symptom_to_ICU	0,03	0,86	TRUE	7.408,00	0,40	0,06	0,47	Accept
Hosp_to_ICU	0,30	0,58	TRUE	7.669,00	0,69	0,03	0,49	Accept
Survival	0,64	0,43	TRUE	8.201,00	0,43	-0,04	0,52	Accept
Intubation	1,58	0,21	TRUE	8.295,00	0,44	-0,05	0,53	Accept

DataSynthesizer hypothesis test results for 100 new synthetic data.

Variable	**W**	**pval**	**equal_var**	**U-val**	**p-val**	**RBC**	**CLES**	**result**
Age	0,23	0,63	TRUE	11.596,00	0,75	0,02	0,49	Accept
Gender	1,76	0,19	TRUE	12.622,00	0,19	-0,07	0,53	Accept
Heart_rate	0,05	0,83	TRUE	11.767,00	0,92	0,01	0,50	Accept
SBP	0,11	0,74	TRUE	11.209,00	0,41	0,05	0,47	Accept

DBP	0,18	0,67	TRUE	10.766,00	0,17	0,09	0,45	Accept
SpO2	0,45	0,50	TRUE	12.395,50	0,48	-0,05	0,52	Accept
Max_SpO2	0,20	0,65	TRUE	11.783,50	0,93	0,01	0,50	Accept
Temperature	0,00	0,97	TRUE	13.279,00	0,07	-0,12	0,56	Accept
BMI	0,04	0,85	TRUE	10.790,00	0,17	0,09	0,46	Accept
pH	0,22	0,64	TRUE	12.561,00	0,36	-0,06	0,53	Accept
PaCO2	0,06	0,81	TRUE	11.512,00	0,67	0,03	0,49	Accept
GCS	1,19	0,28	TRUE	13.728,00	0,00	-0,16	0,58	Reject
APACHE_II	0,46	0,50	TRUE	11.552,00	0,70	0,03	0,49	Accept
SOFA	0,06	0,81	TRUE	11.227,00	0,42	0,05	0,47	Accept
PaO2_FiO2	0,12	0,73	TRUE	12.178,00	0,68	-0,03	0,51	Accept
Creatinine	2,21	0,14	TRUE	11.324,00	0,50	0,04	0,48	Accept
Leucocytes	0,21	0,64	TRUE	11.863,00	0,99	0,00	0,50	Accept
CRP	1,07	0,30	TRUE	12.117,00	0,73	-0,02	0,51	Accept
D_dimer	0,11	0,74	TRUE	10.073,50	0,02	0,15	0,43	Reject
ROX_index	0,03	0,85	TRUE	11.795,00	0,94	0,01	0,50	Accept
Symptom_to_ICU	0,00	0,99	TRUE	11.237,50	0,43	0,05	0,47	Accept
Hosp_to_ICU	2,05	0,15	TRUE	11.505,00	0,65	0,03	0,49	Accept
Survival	0,41	0,53	TRUE	11.551,00	0,52	0,03	0,49	Accept
Intubation	1,01	0,32	TRUE	12.324,00	0,48	-0,04	0,52	Accept

DataSynthesizer hypothesis test results for 150 new synthetic data.

Correlation Matrix Evaluation

Variables	GC	GAN	DS
Age	0,10	0,13	0,11
Gender	0,06	0,13	0,11
Heart_rate	0,10	0,17	0,12
SBP	0,13	0,18	0,10
DBP	0,09	0,12	0,12
SpO2	0,08	0,14	0,10
Max_SpO2	0,11	0,13	0,11
Temperature	0,10	0,15	0,11
BMI	0,10	0,17	0,13
pH	0,10	0,20	0,09
PaCO2	0,11	0,18	0,10
GCS	0,12	0,12	0,09
APACHE_II	0,12	0,21	0,11
SOFA	0,12	0,22	0,08
PaO2_FiO2	0,10	0,17	0,09
Creatinine	0,10	0,17	0,12
Leucocytes	0,11	0,12	0,14

CRP	0,10	0,24	0,11
D_dimer	0,14	0,13	0,11
ROX_index	0,08	0,15	0,08
Symptom_to_ICU	0,13	0,18	0,13
Hosp_to_ICU	0,14	0,14	0,12
Survival	0,13	0,18	0,11
Intubation	0,16	0,20	0,10
Total	0,11	0,16	0,11

Correlation matrix evaluation for 50 new synthesized data

Variables	GC	GAN	DS
Age	0,11	0,15	0,08
Gender	0,10	0,09	0,09
Heart_rate	0,08	0,14	0,08
SBP	0,08	0,11	0,09
DBP	0,09	0,11	0,11
SpO2	0,08	0,14	0,08
Max_SpO2	0,12	0,08	0,10
Temperature	0,09	0,11	0,07
BMI	0,09	0,13	0,05
pH	0,08	0,18	0,10
PaCO2	0,06	0,16	0,09
GCS	0,09	0,13	0,06
APACHE_II	0,09	0,19	0,06
SOFA	0,08	0,22	0,08
PaO2_FiO2	0,11	0,15	0,10
Creatinine	0,09	0,13	0,10
Leucocytes	0,11	0,15	0,12
CRP	0,08	0,19	0,12
D_dimer	0,05	0,10	0,09
ROX_index	0,09	0,14	0,08
Symptom_to_ICU	0,09	0,12	0,09
Hosp_to_ICU	0,12	0,11	0,06
Survival	0,08	0,15	0,09
Intubation	0,09	0,13	0,08
Total	0,09	0,14	0,09

Correlation matrix evaluation for 100 new synthesized data

Variables	GC	GAN	DS
Age	0,07	0,14	0,05
Gender	0,05	0,10	0,05

Heart_rate	0,06	0,13	0,05
SBP	0,07	0,11	0,05
DBP	0,07	0,13	0,04
SpO2	0,07	0,15	0,07
Max_SpO2	0,07	0,11	0,07
Temperature	0,06	0,09	0,05
BMI	0,05	0,08	0,06
pH	0,08	0,19	0,05
PaCO2	0,07	0,16	0,05
GCS	0,07	0,12	0,03
APACHE_II	0,07	0,19	0,04
SOFA	0,06	0,20	0,05
PaO2_FiO2	0,07	0,17	0,06
Creatinine	0,08	0,14	0,07
Leucocytes	0,07	0,12	0,06
CRP	0,07	0,16	0,05
D_dimer	0,08	0,10	0,04
ROX_index	0,07	0,15	0,05
Symptom_to_ICU	0,06	0,13	0,07
Hosp_to_ICU	0,08	0,11	0,09
Survival	0,09	0,13	0,04
Intubation	0,05	0,20	0,05
Total	0,07	0,14	0,05

Correlation matrix evaluation for 150 new synthesized data