EVALUATING PRIVACY OF SYNTHETIC DATA THROUGH METRICS

Bachelorarbeit zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.) im Fach Informatik

Frei Universität zu Berlin
Fachbereich Mathematik und Informatik
Institut für Informatik

DANIEL SOSNOVCHYK

Matrikelnummer: 4870394

daniel.sosnovchyk@zedat.fu-berlin.de

10.03.1995, Kiew

23.August 2021

Betreuung:

Prof. Dr. Marian Margraf
M.Sc Franziska Boenisch

Daniel Sosnovchyk: *Evaluating privacy of synthetic data through metrics*, Frei Universität zu Berlin, 23.August 2021

SELBSTÄNDIGKEITSERKLÄRUNG

Ich erkläre ausdrücklich, dass es sich bei der von mir eingereichten schriftlichen Arbeit mit dem Titel

Evaluating privacy of synthetic data through metrics

um eine von mir selbst und ohne unerlaubte Beihilfe verfasste Originalarbeit handelt. Ich bestätige überdies, dass die Arbeit als Ganze oder in Teilen nicht zur Abgeltung anderer Studienleistungen eingereicht worden ist. Ich erkläre ausdrücklich, dass ich sämtliche in der oben genannten Arbeit enthaltenen Bezüge auf fremde Quellen (einschließlich Tabellen, Grafiken u. Ä.) als solche kenntlich gemacht habe. Insbesondere bestätige ich, dass ich nach bestem Wissen sowohl bei wörtlich übernommenen Aussagen (Zitaten) als auch bei in eigenen Worten wiedergegebenen Aussagen anderer Autorinnen oder Autoren (Paraphrasen) die Urheberschaft angegeben habe. Ich nehme zur Kenntnis, dass Arbeiten, welche die Grundsätze der Selbständigkeitserklärung verletzen – insbesondere solche, die Zitate oder Paraphrasen ohne Herkunftsangaben enthalten –, als Plagiat betrachtet werden können. Ich bestätige mit meiner Unterschrift die Richtigkeit dieser Angaben.

Berlin, 23.August 2021

Daniel Sosnovchyk

Daniel Sosnovchyk

ABSTRACT

Making anonymized datasets public is proven to have inherent privacy risks. One option to avoid the privacy risks of publishing data, is not to disclose real data but generated data. Generation algorithms learn, as a simplification, the joint probability distributions of the real data and create a new dataset, called synthetic dataset. Though it is believed that disclosing synthetic data has little privacy risk, it is possible to overfit the real data. This would mean that the generated synthetic data is too close to the real data. If the synthetic data is too similar, it is possible to map the synthetic data records to the real ones, causing a privacy risk for individuals in the dataset. For this reason, the privacy of the generated data needs to be assessed. In this work, synthetic data will be generated and be analyzed using various metrics. One approach is to measure the similarity between the real and synthetic dataset. If the datasets are too close, the privacy is assumed to be low. Some similarity approaches used in analysis of the data are implemented in this work. Another approach will be created that estimates the probability of an adversary linking synthetic data to real data. Adding this metric will give a more rounded approach to estimate the degree of privacy of the generated data.

KURZFASSUNG

Die Veröffentlichung anonymisierter Datensätze birgt nachweislich Risiken für den Schutz der Privatsphäre in sich. Eine Möglichkeit, die mit der Veröffentlichung von Daten verbundenen Risiken für den Schutz der Privatsphäre zu vermeiden, besteht darin, keine echten Daten, sondern generierte Daten zu veröffentlichen. Generierungsalgorithmen lernen die gemeinsamen Wahrscheinlichkeitsverteilungen der realen Daten und erstellen einen neuen Datensatz, genannt synthetischer Datensatz. Obwohl man davon ausgeht, dass die Offenlegung synthetischer Daten nur ein geringes Risiko für den Schutz der Privatsphäre birgt, ist es möglich, die realen Daten zu over-fitten. Dies würde bedeuten, dass die erzeugten synthetischen Daten den realen Daten zu ähnlich sind. Wenn die synthetischen Daten zu ähnlich sind, ist es möglich, die synthetischen Datensätze den realen Datensätzen zuzuordnen, was ein Risiko für die Privatsphäre der Personen im Datensatz darstellt. Aus diesem Grund muss die Privatsphäre der generierten Daten bewertet werden. In dieser Arbeit werden synthetische Daten generiert und anhand verschiedener Metriken analysiert. Ein Ansatz besteht darin, die Ähnlichkeit zwischen dem realen und dem synthetischen Datensatz zu messen. Wenn die Datensätze zu ähnlich sind, wird davon ausgegangen, dass die Privatsphäre gering ist. Einige Ähnlichkeitsansätze, die bei der Analyse der Daten verwendet werden, werden in dieser Arbeit implementiert. Es wird ein weiterer Ansatz entwickelt, der die Wahrscheinlichkeit schätzt, mit der ein Adversary synthetische Daten mit realen Daten verknüpft. Durch die Hinzufügung dieser Metrik wird ein vollständigerer Ansatz zur Abschätzung des Grades der Privatsphäre der generierten Daten geschaffen.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

With increasing digitization, in the medical field, public administration, and private companies, more and more data is being collected. The collected data could be leveraged to answer many domain-specific questions. For some questions, the amount of data that one single company or medical institution can collect is not enough. Or the company that is collecting the data does need to outsource the data analysis to a third party. The problem with sharing information, especially when dealing with information of individuals is with regards to protecting the privacy of individuals. Under the EU General Data Protection Regulation, personal data has to be rendered anonymous in such a way that the individuals are no longer identifiable [1]. So making anonymized data available to researchers would benefit the holder of the data. In the medical fields, some advancements can be done that benefit everyone. There is an incentive to share the collected data or even to make it public. The incentive is not coming from the law alone, since disclosure of privacy of costumers in private companies would do considerate damage to that companies image. For privacy violations coming from the medical field, it could mean that not as many people are willing to participate in studies if they think that sensitive information can be disclosed.

One approach to render personal data anonymous is a simple redaction of identifying information or pseudonymization. These approaches are shown to be not enough to prevent an individual from reidentification. A few facts can be combined to isolate an individual [2]. Publicly available information can be also used to re-identify an individual from an anonymized data set [2].

One solution to this problem is the use of synthetic data. Synthetic data is created from a real data set and shares the distributions found in the real data. Generated correctly, it should look real and have the statistical properties of the original data set [3].

It is generally believed that the privacy risk of releasing synthetic datasets is negligible. "Identification of units and their sensitive data from synthetic samples is almost impossible" is stated in [4].

There are many methods to create synthetic data. To create synthetic data that looks real, the parameters of the generation methods need to be set optimally for the original data set that will be used to train the model of the generation algorithm. The algorithm could overfit the synthetic model to the real data using wrong parameters for the creation [5]. It is also possible that the synthetic data generation method has a bug that creates synthetic data that is too close to the real data.

If the synthetic data is too close to the real data, an adversary would be able to look up a combination of publicly known identifying information or quasi-identifiers of an individual in the synthetic dataset and find out sensitive information too close to the information of the real dataset.

In research related to synthetic data, the created synthetic data is not tested rigorously for privacy. This thesis has the goal to apply available privacy metrics, as well as to create privacy measure, specific to synthetic data, to assess if the generated synthetic data can be considered private or not.

1.1 STRUCTURE OF THIS THESIS

At first I will introduce the research I did regarding the topic of synthetic data and privacy in chapter 2. There I will give an overview of common terms that are used in this field in section 2.1 and show common analysis techniques that analyze synthetic data in section 2.2

After that I will show how I created the synthetic data for this thesis in chapter 3, which dataset I use section 3.1 and which methods I chose to create synthetic data section 3.3.

In chapter 4 I will evaluate the synthetic data I generated for privacy.

Finally, I will summarize my findings and give an overview of the collected information in chapter 5.

2 RELATED WORK

To discuss the topic in the context of the body of works linked to privacy and synthetic data, first an introduction to central term that are used in these scientific topics. Then I will go over relevant papers that that discuss synthetic data and give an overview how synthetic data is analyzed.

2.1 USED TERMINOLOGY

In this section, I will explain the basis terms used in this thesis.

Synthetic data
Synthetic data is data that was not directly measured. In the context of the thesis, it is data that is generated by using a real dataset. Synthetic data generation algorithm learn the joint probability distributions of the real data and generate a new dataset, the synthetic dataset, with similar probability distributions. In this work, synthetic data is generated to preserve the privacy of individuals in the real dataset. In this thesis only tabular data is looked at but there are more fields of application for synthetic data, like to synthesize pictures (e.g. for analysing CT scans in the medical field) or 3D models (e.g. for training recognition of objects). Use cases for synthesizing data can be [5]:

- To enlarge the real dataset if more data is required for machine learning purposes.
- Creating datasets where no real data was previously collected and there is only a model for testing theories.
- Creating labeled data for machine learning. Labeling of data is expensive because it needs manual labor. Creating data where we know the label is a lot cheaper and a lot more variations can be created.

Privacy
Privacy means that the individuals are no longer unidentifiable in accordance to the EU law, mentioned in chapter 1. Privacy is discussed around data of individuals in datasets that contains information that should not be public, e.g. medical data. Individual refers, as used in the thesis, to a natural person but also to any entity (like company or institution) that has data that is sensitive.

Differential Privacy
Differential privacy is a mathematical definition of privacy. While differential privacy is not used in the privacy metrics, it is one central concept for privacy. Some synthetic data generations do provide an option to create differential private synthetic data one such dataset will be created in this thesis. To explain differential

privacy, first we need to understand when privacy of an individual in a dataset is considered to be violated. Lets assume, a study is conducted that wants to find a link between smoking and lung cancer. In this study, data from a number of smokers and non-smokers is collected. Lets assume, an individual is known to be a smoker and partakes in this study. If the study confirms that the connection of smoking and cancer exists, the individual, under the above definition, would consider his privacy violated. It is learned that he (as other smokers) has a higher risk of cancer, compared to non-smokers. So a new information is learned about the individual. Following the definition that privacy is violated if something new is learned about an individual, data analysis will not be possible. As this would be an issue, a softer definition of privacy violation is needed. If the individual would be replaced by another smoking member of the population, the same conclusion about the link can be made. The formal definition of differential privacy follows, the outcome of any dataset analysis should be the independent of any single individual in the dataset. Differential privacy is regarded as the strongest methods to ensure privacy of individuals, being used by Apple and Google to secure the privacy of their clients [6] [7]

Adversary
The adversary is a data analyst that has the goal to compromise data privacy to learn new information about the individuals in the dataset [8].

Identifying information
Identifying information is a data value in the dataset that can be used to deduce the identity of the individual, e.g. the name of the person [9].

Sensitive values
Sensitive values is classified information that needs to remain anonymous, e.g. the medical records of an individual [9].

Quasi-identifiers
Quasi-identifiers are values that can not be used to identify an individual from a dataset but can be used in combination with other quasi identifier to do exactly that. Data researches showed that combining the quasi identifiers "zip code", "gender" and "date of birth", around 87% of Americans can be clearly identified from a census data dataset [9].

Equivalence class
An equivalence class is a dataset is a collection of all row that share the same semi-identifiers [9].

2.2 PAPERS ON THE TOPIC OF SYNTHETIC DATA

To look how synthetic data is being analysed, some papers that deal with it were looked into first. Algorithms that generate private synthetic data are introduced in [10], [11] and [5]. In these works, the analysis of privacy is done while viewing the pseudo-code of the algorithm.

All papers analyzed the output of the algorithm similarly, while the structure of the analysis was clearest in [10]. Therefore, the next part will mostly use the terminology and description of said paper.

The evaluation of the output of the algorithm is split into four distinct logical parts.

1. **Statistical Measures**
   The idea of the statistical measures is to evaluate how close the statistical properties of the synthetic data are to the statistical properties of the original dataset. If the statistical properties of the datasets are very different it would mean that the utility of the synthetic dataset will be likely low. If the statistical properties are too close, it would suggest a privacy risk.

   The statistical evaluation was performed on the level of the generative model and also by comparing the synthetic data to real data. The comparison of synthetic data is to real data is the more interesting technique for this theses because the generative model is not disclosed for evaluation. This comparison was done by looking at the probability distributions of the synthetic data and the real data, specifically the distribution of each attribute and also every pair of attributes. The difference in distributions for synthetic and real datasets is measured by using the statistical distance (also called the total variation distance). The difference can be shown in a box-and-whiskers plot.
   This measures the utility of the synthetic data is important in the analysis of the synthetic data.

2. **Machine Learning Measures**
   The idea of machine learning measures is to use real dataset and the synthetic dataset for machine learning tasks. If the algorithm trained on synthetic data yields similar results to the one that is trained on real data, the utility of the synthetic data is high. The specific task described in [10] was a classification task. The agreement rate was measured that indicates if the two algorithms give the same of different outcomes. The analysis of this measure would be too complicated for this work.

3. **Distinguishing Game**
   The distinguishing game is the idea that a participant has to guess if a record is from the real dataset of the synthetic one. This shows if the synthetic data looks like real data or not. The participants that are supposed to play the game are classifiers, Random Forest, and Classification Tree. They are given a part of the synthetic data to train. The part of the data that was not used for training, is used for the game. Implementation of the distinguishing game would also be beyond the scope of this thesis.

4. **Performance Measures**
   The performance measures show how long it takes to create the generative model as well as the generation of synthetic data. Because I will treat the code as a black box, only the duration of the generation can be looked upon in this work.

The evaluation of the synthetic data in these papers gives a measurement of the quality of the synthetic data in regards to utility and how much the synthetics looks like the real data. An evaluation of the degree of privacy of the data is missing. Doing the analysis of an algorithm is viable if there is insight into the algorithm. There are many services and tools that promise generating differential private data. This promises can not be verified because the algorithm that generates the synthetic data acts like a black box. Also, even if the code is visible, it is not guaranteed that because of coding errors or poor implementation of the pseudo code, the implemented algorithm still gives the same privacy guarantees. To do the evaluation of privacy on the level of the pseudo code alone is not enough. There need to be measures that can give an assessment of privacy of the algorithm on the side of the output synthetic data.

3 FROM REAL DATA TO SYNTHETIC DATA

The creation of synthetic data and the analysis of it, requires a dataset to train the algorithm. The meaning of the values in the data set is not important for the context of this work. For this reason, the chosen original data will undergo some data transformation techniques to make it easier to work with the algorithms that will be introduced in section chapter 4. If the original values matter, it would be possible to perform the inverse of the transformations to get meaningful data in these cases.

Three different methods for creating synthetic data will be used. These methods were already implemented. Creating datasets using different methods should provide differences in synthetic data utility, which should come up in the analysis part in section chapter 4.

3.1 DATASET

The dataset that is used for this work is the Pima Indians Diabetes dataset from Kaggle [12]. This dataset has 768 rows and consists of numerical values that are mapped to a target variable, the outcome, as 0 or 1.

The data was collected from real patients, all female, at least 21 years old with Pima Indian descent. The dataset comes from the National Institute of Diabetes and Digestive and Kidney Diseases and is used to predict if a patient has diabetes [13]. The meaning of the column names used in the dataset can be seen in table Table 3.1. Some rows of the original dataset can be seen in table Table A.0.

The distribution of values from the original dataset can be seen in figure Figure 3.1.1. Upon closer inspection of the data, there are measurements with a value of zero which are not possible in in real life. Glucose amount in the blood, blood pressure, skin thickness, amount of insulin, and BMI had missing values which were replaced with zeroes. These zeros must be deleted of the dataset otherwise the algorithm would see them as measured values which they are not. Especially the column, "SkinThickness", has a lot of missing values as seen in the Figure 3.1.1 The distributions on the data in columns "Glucose", "BloodPressure", "SkinThickness" and "BMI", without the zeros, is Gaussian like, which is needed for the synthetic data generation in section 3.3 (otherwise, the synthesized data would be very different form the training data).

A solution for filling the gaps is needed. This will be addressed in section 3.2.

3.2 PREPROCESSING OF THE DATA

In the best case, we would have the dataset in a state where it can easily be used by all coming algorithms. Some of the algorithms require data to be in a certain form. The VAEs and GANs algorithms from section 3.3 require the data distributions to

Table 3.1: Meaning of columns values in the Pima Indians Diabetes dataset, in the first column, the name of collected variables is mentioned. In the second column, the is a short explanation of the measured values.

Name of the column in the dataset	**Explanation of the measured variables**
Pregnancies	*Number* of times pregnant
Glucose	Plasma glucose concentration a 2 hours in an oral glucose tolerance test
BloodPressure	Diastolic blood pressure in $mmHg$
SkinThickness	Triceps skin fold thickness in mm
Insulin	2-Hour serum insulin in $mu\frac{U}{ml}$
BMI	Body mass index
DiabetesPedigreeFunction	Diabetes pedigree function calculation from the original paper
Age	Age in *years*
Outcome	Target variable 0 or 1 for having no diabetes or having diabetes

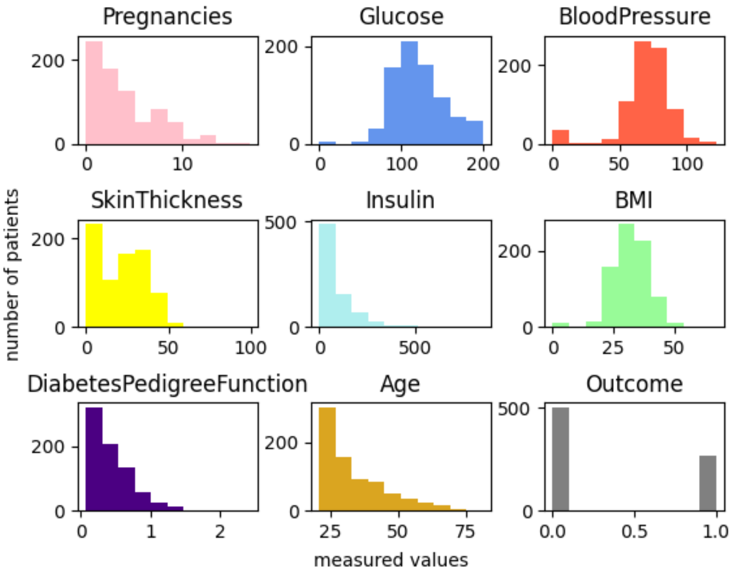

Figure 3.1.1: Histograms of all columns of the original data

be Gaussian or close to Gaussian. For this reason, data preprocessing is required. The reasons for doing each step will be mentioned in the below descriptions of the data preprocessing steps.

The dataset created after the preprocessing will be called **train_data**.

3.2.1 *Removing the missing values with the KNN impute algorithm*

There are different ways to deal with the missing values as zeroes. It is also possible to delete the out-of-place zeroes and work with the gaps in data. This would be valid, but it would make the analysis and application of algorithms more difficult. One other way would be to remove all rows with missing values. But this would result in a dataset that would be too small for synthetic data generation, so this also is not an option. The best strategy for the given dataset, in dealing with missing values, is data imputation. Data imputation means to fill in the missing values. There are many data imputation strategies that can be chosen. There are simple methods like replacing a missing value with the mean of the column to more sophisticated machine learning methods. The method that was chosen for this data is the KNN impute [14] algorithm. KNN stands here for k-Nearest Neighbors, with k being a nonnegative integer greater than 1. To replace a missing value, the algorithm considers k rows that are similar. The similarity is measured by the euclidean distance to other rows.

The paper [14] shows that the KNN method is more accurate than taking the average value of a column or leaving the zeroes in place. From the results in the linked paper, the KNN method was around 6% better than the the average method and around 7.5% better than replacing missing values with zeroes. The paper [14] also shown that the replaces values had $6 - 26\%$ average deviation from the true values, depending on the type of data and fraction of values missing. From the experiments done in the paper, k value for KNN was chosen to be 17. This number was in the middle of the k values that produced the best results in their experiments. As seen in the figure 3.2.1, the data imputation helped some column values to look more Gaussian. Because the euclidean distance was used as a metric here, data scaling was applied first.

3.2.2 *Transforming Data to be more Gaussian-like*

The coming data generation algorithms in section 3.3 work best if the data is Gaussian or Gaussian-like [15] [16] as the used algorithms fit a gaussian function over the data. The geometric distributions have to be altered, if not, the synthesized data would be very different than the training data. For this, Yeo-Johnson transformation was used on the data. As the theory and the mathematical description of the transformation is fairly complicated and is not focus of the thesis, the description of the method is explained in detail in [17] and will not be further explained here.

After the transformation, as seen in the histogram 3.2.2, the previously geometrical distribution for "DiabetesPedigreeFunction", "Age" and "Pregnancies" look more Gaussian as seen in Figure 3.2.2.

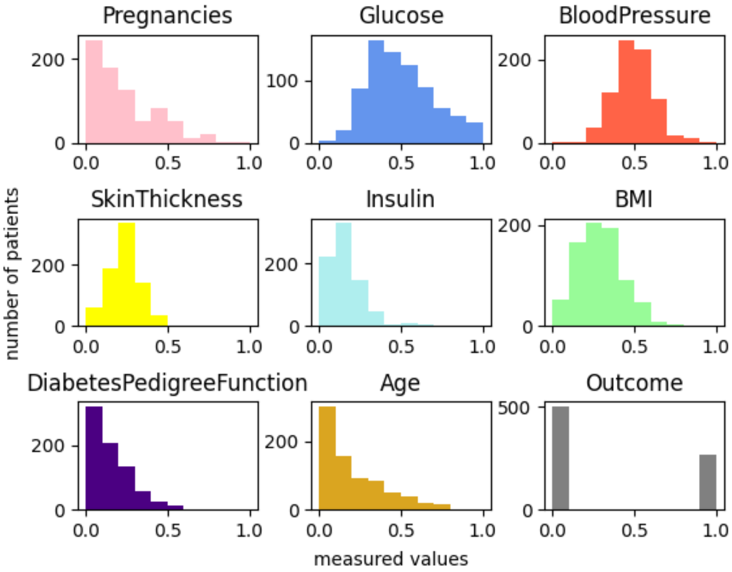

Figure 3.2.1: Histograms of data after imputation of missing values. As can be seen in the "SkinThickness" column, the peak at the measurement zero is gone.

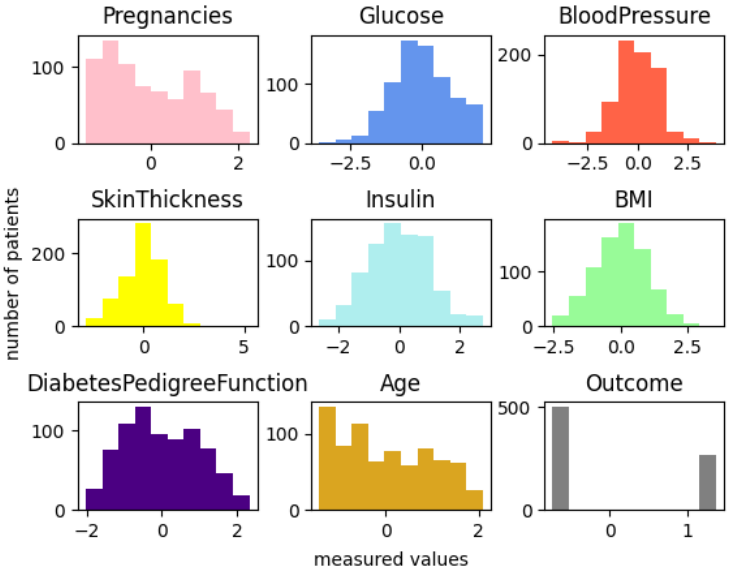

Figure 3.2.2: Histograms of data after the transformation. The data distributions are more Gaussian.

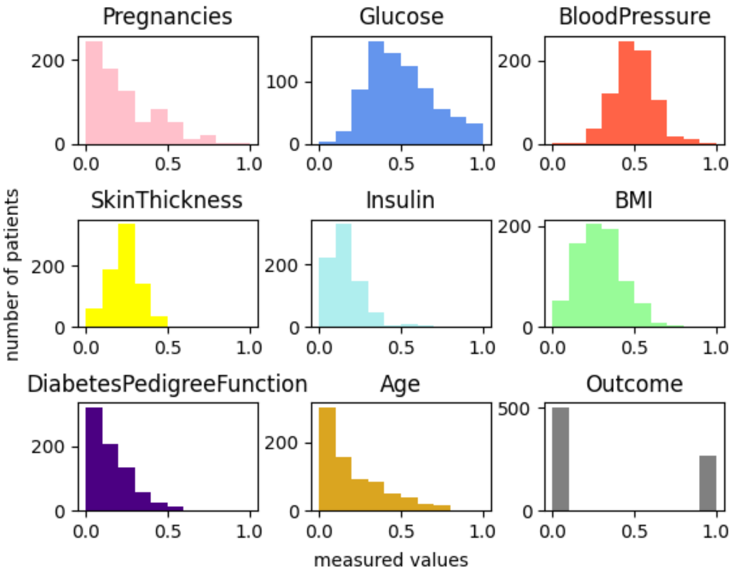

Figure 3.2.3: Histograms of data after feature scaling. This is the final stage of preprocessing. This dataset will be used for the generation of synthetic data.

3.2.3 *Scaling*

The euclidean distance, as a measure, like in 3.2.1 or 4.1.6, is very scale dependant. This means that the results can vary enormously depending on the ranges of values [18]. One standard method to deal with it is a Min-Max scaling. This method scales the values per column so the values are uniformly between the min value, scaled to 0 and the max value, scaled to 1. Because the values, in between 0 and 1, are scaled uniformly, the outliers in the dataset are preserved. Preserving outliers in important. Outliers are interesting because they have more unique properties that make them stand out. The privacy of individuals with outstanding values in the dataset needs to be also preserved. Finding out if the privacy of outliers in preserved in interesting in the analysis. In Figure 3.2.3 can be seen, all values are between 0 and 1 . At this stage we got our training data, that will be used as the basis for all further steps.

3.3 Generating synthetic data

The synthetic data generation algorithms are not the focus of this work but I will describe the methods briefly to give a basic idea how the data is generated. The created data will have the same size as the training data as it is a requirement for most algorithms in chapter 4.

3.3.1 *VAEs - Variational Autoencoders*

Normal Autoencoders, see Figure 3.3.1, are a neural network method that represent the input in a compressed way. The input goes through the encoder part of

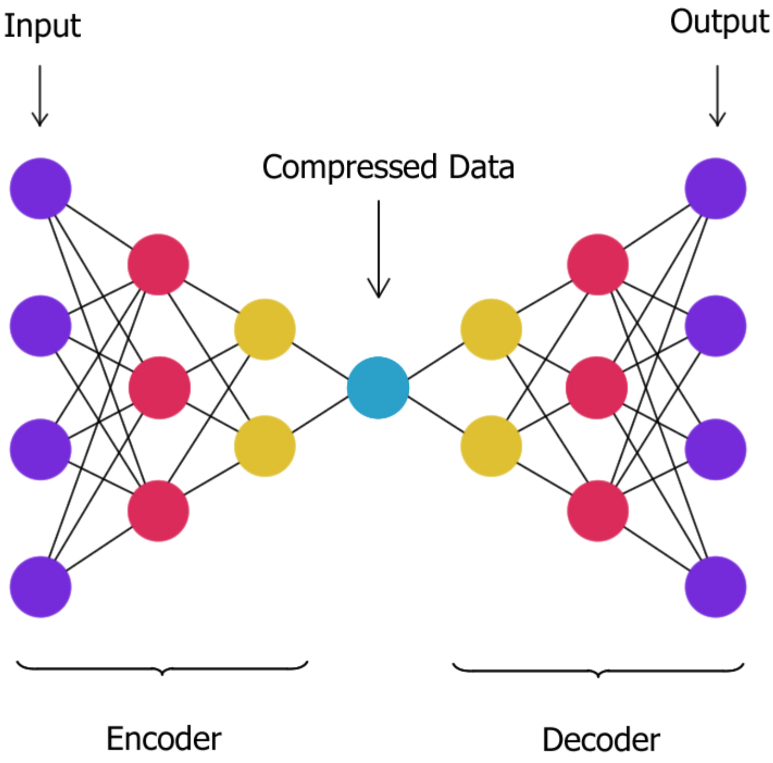

Figure 3.3.1: Schematic of a basic autoencoder. It is a neural network that runs input data though an encoder to a lower dimensional space. The decoder then reconstructs the data to the same dimensionality it had before. The layers with the same colors have the same amount of neurons.

the neural network. The number of neurons in each layer is shrinking until it is compressed. The data is then sent back through the decoder that reconstructs the data back. The number of neurons in the input layer and the output layer is the same, meaning that ideally the output is the same as the input, or very similar [19]. In a normal Autoencoder, all steps are done deterministic, the data is defined by deterministic mapping. Deterministic means that for a for a particular input and a defined neural network, all steps will be done in the same way. In the Variational Autoencoders (VAE), however the encoder and decoder work probabilistic [20].

The implementation of the method to generate tabular data with GANs was taken from [15]. The implementation allows to enable an option that the generated data is differentially private. Thus, two synthetic datasets are created using this method, **vae_public** for the nonprivate data and **vae_private** for the differentially private data.

The algorithm of the implementation is not working as documented. In the instructions of the method, only *int* and *categorical* values were used, not floats as in the dataset I using. The first issue was that the algorithm did not work at all with the *train_data* dataset. The code does not work if no columns contain strings. To solve this issue, 0 and 1 values in the "Outcome" column are changed to "true" and "false" ("t" and "f" did not work). The second issue were the floats. Floats were just rounded off. In the case of the scaled data set where all values are between 0 and 1, only these two values were contained in the dataset from this point on. To solve this issue, I multiplied all values with one million to carry over a good amount post decimal positions, generated data, and divided all values with one million to be back on the same scale.

3.3.2 *Gretel AI*

Gretel is a service with which synthetic data can be created. Their core synthetic data generation library in on GitHub [21][22]. The exact algorithms and methods that they are using are not described. The dataset created with with service is **gretel_public**. It is also possible to create data that is differentially private. The algorithm cannot create a synthetic data set that is only 768 rows, Gretel's recommendation is to use a data set above 5000 rows. No dataset with differential privacy was created with Gretel for this reason. The creation of synthetic data through the website was very easy, giving statistical measures about the data utility after the creation.

3.3.3 *GANs - Generative adversarial networks*

Generative adversarial networks (GANs) is a machine learning technique characterized by a pair of networks competing with each other [23]. On one side there is a generator who is generating data, in the beginning only from random noise. On the other side is the discriminator. The generator himself does not have access to real data. The discriminator trains to distinguish the generated data from real data and gives feedback to the generator. The generator learns to create better looking data from the feedback of the discriminator. Both parts train at the same time and are competing with each other. [23]. The implementation of the method to generate tabular data with GANs was taken from [16]. The method is very basic and does not provide an option to create private data. The dataset created with this method will be called **gans_public**. The generation using this method required to try different parameters for the creation, like number of epochs for training or size of the latent space, to get somewhat usable data, which took some time.

3.3.4 *A control dataset*

In section chapter 4 the privacy of generated synthetic data will be measured. It is possible that the privacy of the generated data is good across the board. The reasons could be that the synthetic data is very dissimilar to the training data and private because it has to overlap or in the case that the data is similar but can't be used to recognize someone in the dataset. Since the idea that measuring the data after the generation, e.g. in case there is a bug in the code, it would be valuable to have a dataset where it is known that the privacy is compromised. For this reason, an additional dataset will be created. To create this control dataset, the **gretel_public** will be used as a baseline but 1% of its rows will be replace by rows from the training dataset. Because if is easy to detect that some rows are the same, the copied values will be changed in the ±1% range to make the rows very similar but distinct from the rows in the training data set. This dataset will be called **gretel_corrupted** in the following work.

The datasets will be analyzed in the following chapter.

4 EVALUATION OF GENERATED DATA

In this chapter the previously obtained datasets will be analyzed. In the first part, section 4.1, I will go over data similarity measures that were found in papers [5][24][10][9]. In the second part, section 4.2, I will argue why the data similarity measures do not give a valuable estimate of the privacy of data.

4.1 DATA SIMILARITY

Measures of data similarity are used heavily as methods assessing the quality of synthetic data in research. There are more similarity measures that are available but I will go over some that can be easily implemented. High data similarity is a double edged sword. Without high data similarity, the generated data is not usable as it does not have the statistical properties of the original data. With low resemblance of the generated data to the original data, the privacy is high, as it would be not possible to link data from the synthetic dataset to the real dataset. Having a high similarity of above 95%, as stated in [5], overfitting of the data could have happened. So, an adversary could link the synthetic data with higher precision to real data.

4.1.1 *Basic metrics*

Some of the basic statistical properties are the mean of values and standard deviation. In [24], it is suggested the differences between the mean and the standard deviation can be calculated by the following method:

$$\text{Similarity} = \rho_{\text{spearmany}}(\langle \text{mean}(R), \text{std}(R) \rangle, \langle \text{mean}(F), \text{std}(F) \rangle) \tag{4.1.1}$$

where $\langle A, B \rangle$ means the concatenation of the lists A and B. $\text{mean}(A)$ means the list of the means over all columns of A and $\text{std}(A)$ is the standard deviation of the column values. ρ_{spearman} is the Spearman's correlation coefficient that is a measure of correlation between two lists, further explanation of the ρ_{spearman} can be found in [24].

The informational value of this measure is limited and should not be used on its on to make a statement about the privacy [24]. If combined with other data utility measures, it can be valuable to have these results as a reference. As seen in Table 4.1.1, the means and standard deviations of the **gretel_public** and the **gretel_corrupted** datasets are almost the same as the ones in the training dataset. Both VAE synthetic datasets, **vae_public** and **vae_public**, follow with high similarity of 0.95 and 0.91 respectably. The means and standard deviation of the column values of the **gans_public** are, in comparison to other datasets, very dissimilar to the ones of the training dataset.

Table 4.1.1: Examination of the similarity of the mean and standard deviations of each column in the synthetic dataset and the training dataset

Dataset	Correlation measure of mean and standard deviation
gans_public	0.7153
vae_public	0.9526
vae_private	0.9109
gretel_public	0.9912
gretel_corrupted	0.9918

4.1.2 *Hellinger distance*

The hellinger distance is a measure of the distance between two probability distributions, used for assessment of the utilty of synthetic data [5]. With $P = (p_1, ..., p_x)$ and $Q = (q_1, ..., q_x)$ being discrete discrete probability distributions, the definition of the hellinger distance is the following:

$$dist_{\text{hellinger}} = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{x} (\sqrt{p_i} - \sqrt{q_i})^2} \tag{4.1.2}$$

Table 4.1.2: Distance between the distributions measured using the hellinger distance. 0 means no difference and 1 mean maximum difference.

	gans _public	vae _public	vae _private	gretel _public	gretel _corrupted
Pregnancies	0.4300	0.6830	0.2567	0.1284	0.1169
Glucose	0.2866	0.4674	0.2935	0.1269	0.1289
BloodPressure	0.3523	0.5402	0.2388	0.0810	0.0794
SkinThickness	0.4099	0.3123	0.3484	0.0954	0.0983
Insulin	0.255	0.5653	0.3864	0.0597	0.0569
BMI	0.2844	0.5358	0.1229	0.1178	0.1171
DPF	0.2649	0.6284	0.4303	0.2288	0.2192
Age	0.483	0.6978	0.4802	0.0709	0.0715
Outcome	0.0019	0	0	0.0019	0.0028
Mean	0.3076	0.4922	0.2841	0.1012	0.0990

The distance returns values between 0 and 1 with 0 meaning minimal distance (maximum similarity) and 1 being the maximum distance (no similarity). The similarity to the training dataset, see Table 4.1.2, is highest for the **gretel_public** and **gretel_corrupted** dataset. In [5] it is stated that the difference closer than 5% would be considered as too close. The difference of about 10% is fine. The **gans_public** and **vae_private** follow with a distance of 0.3, which is a big step in comparison. The distance of **vae_public** is highest with 0.49.

4.1.3 *Total variation distance*

The total variation distance (also known as statistical distance) is also a measure of the distance between two probability distributions. It was used as a measure of utility in [10]. With $P = (p_1, ..., p_x)$ and $Q = (q_1, ..., q_x)$ being discrete probability distributions, the definition of the total variation distance, for sets X that are countable, is the following:

$$\text{dis}_{\text{tvd}}(P, Q) = \frac{1}{2} \sum_{x \in X} |P(x) - Q(x)| \tag{4.1.3}$$

The total variation distance is also related to the Hellinger distance as follows:

$$dist^2_{\text{hellinger}}(P, Q) \leq dist_{tvd}(P, Q) \leq \sqrt{2}\, dist_{\text{ hellinger}}(P, Q) \tag{4.1.4}$$

Table 4.1.3: Distance between the distributions measured using the total variation distance. Bigger numbers suggest more difference.

	gans _public	vae _public	vae _private	gretel _public	gretel _corrupted
Pregnancies	0.4388	0.7513	0.2239	0.1236	0.1184
Glucose	0.3567	0.4322	0.3723	0.156	0.1549
BloodPressure	0.3196	0.5338	0.2109	0.0598	0.0585
SkinThickness	0.3255	0.2721	0.3658	0.0872	0.0924
Insulin	0.2955	0.6002	0.3945	0.0611	0.0559
BMI	0.2382	0.5468	0.1158	0.0963	0.1028
DPF	0.2734	0.6901	0.4010	0.2630	0.25
Age	0.4192	0.7773	0.4101	0.0859	0.0865
Outcome	0.0026	0	0	0.0026	0.0039
Mean	0.2966	0.5115	0.2771	0.1040	0.1026

As the total variation distance is related to the hellinger distance, the distance seen in Table 4.1.3, are very similar to the distances in Table 4.1.2.

4.1.4 *Mean distance to nearest neighbor*

As a by-product of subsection 4.1.6, we have a list of distances, seen inTable 4.1.4, that can be used to calculate the mean of the distance to the nearest neighbor. For every row in the real dataset, the distance of the closest synthetic row was calculated. The mean distance to the closest neighbor can be seen in Table 4.1.4. Low numbers mean that the synthetic dataset is closer to the real dataset.

Using this metric, the difference to the real dataset is lowest in **vae_public**, **gretel_public** and **gretel_public** dataset with distances of 0, 06, 0.083 and 0, 080. The difference in datasets is higher in **vae_private** and **gans_public** with distances of 0.11 and 0.14.

Table 4.1.4: For every row in the real dataset, the distance of the closest synthetic row was calculated. The mean distance to the closest neighbor can be seen. Low numbers mean that the synthetic dataset is closer to the real dataset.

Dataset	Mean of the distances to the nearest neighbor
gans_public	0.1405
vae_public	0.0612
vae_private	0.1103
gretel_public	0.0838
gretel_corrupted	0.0801

4.1.5 *Relative Entropy*

Relative Entropy, also known as Kullback-Leibler divergence D_K, is a measure taken from [9]. Relative Entropy measures the distance between two probability distributions. The original use case where this metric is applied, is when there is a true distribution X^* and an adversary estimate X. The metric gives the amount of probabilistic information that the adversary gained. In the context of synthetic data, relative entropy indicates how much the synthetic dataset differs from the training dataset.

With $P = (p_1, ..., p_x)$ and $Q = (q_1, ..., q_x)$ being discrete probability distributions, the definition of the Kullback-Leibler divergence is the following:

$$\text{dist}_{\text{KL}}(P||Q) = H(P, Q) - H(P) \tag{4.1.5}$$

$priv_{CE}$ is the cross entropy, results seen in Table 4.1.5,

$$H(P, Q) = -\sum_i P_i \ \log_2(Q_i) \tag{4.1.6}$$

$H(P)$ is the entropy of P, results seen in Table 4.1.6:

$$H(P) = -\sum_i p_i \ \log_2(p_i) \tag{4.1.7}$$

The amount of difference measured using relative entropy can be seen in Table 4.1.7. The results are again similar to the ones found using hellinger distance.

4.1.6 *Rows that are too close to the original dataset*

One obvious red flag would be if the same rows could be found both in the synthetic dataset and the training dataset. This could indicate a mistake with the parameters before the training or with an issue with the synthetic data generation algorithm. The only way that this should be possible is if a particular row in the training data set is copied several times. One can assume, this is not ideal (clarified in section 4.2) for privacy if the synthetic dataset contains rows from the original data. If the issue comes up because of the training data, it could be

Table 4.1.5: In this table the cross entropy of columns between the original distribution and the synthetic distribution, calculated using Equation 4.1.6. This data is needed to calculate the relative entropy.

	gans _public	vae _public	vae _private	gretel _public	gretel _corrupted
Pregnancies	4.2762	1.961	2.8775	3.2719	3.2555
Glucose	3.4442	1.6502	3.3246	2.9159	2.9198
BloodPressure	2.8068	0.7917	2.6803	2.3195	2.3182
SkinThickness	2.2365	1.8537	2.7465	2.3475	2.3509
Insulin	3.4182	2.8563	3.375	2.9428	2.941
BMI	2.4289	2.8091	2.8124	2.752	2.7508
DPF	3.3291	1.9359	3.5784	3.4871	3.4592
Age	3.0817	2.6348	2.2549	3.2512	3.2535
Outcome	0.9332	0.9331	0.9331	0.9332	0.9332

a good idea to remove the duplicate rows before generating synthetic data. The number of rows, that were found in the training dataset, that are identical to rows in the synthetic dataset, was 0 for every synthetic data set.

Though, the number of rows that are exactly the same is zero for every generated synthetic dataset I have, one can imagine a very similar red flag. The rows are not the same but the values of a row in the synthetic dataset only varies a little bit from a particular row in the training dataset. For example, a row can be found that is the same in the training dataset as in the synthetic data set with the exception of one value, which is slightly bigger in one dataset. The implication to privacy is very similar to the case where the rows are exactly the same.

To test if there are rows very similar in both datasets, for every row in the synthetic data set, the most similar row will be found. The similarity (distance) metric that was chosen is the euclidean distance. This is an easy to use metric because all values in our dataset are numerical. The basic equation can be seen in Equation 4.1.8 with p and q as points and d as the dimensionality of the data [25].

$$\mathrm{dist}(p,q) = \sqrt{(p_1-q_1)^2+(p_2-q_2)^2+(p_3-q_3)^2+...+(p_d-q_d)^2}, \tag{4.1.8}$$

For every row of the synthetic dataset we measure the euclidean distance to all rows in the training dataset. The one with the smallest distance is deemed as the most similar. This method can be sped up by building a data structure for the training data where we can queue only the rows from the synthetic dataset. This would speed up the algorithm for dataset with bigger amount of values. The data structures that can be used here are the KDTree and BallTree binary trees. The data structures are explained in [26]. The distance that we calculated is not intuitive and does not have any meaning by its own apart as being used to compare distances. To give a more intuitive meaning to the distance, we can scale the distance to be between 0 and 1, with 0 meaning that the distance is minimal (also means that the compared rows are the same) and 1 as the maximal

Table 4.1.6: In this table the entropy of each column is shows. It was calculated using Equation 4.1.7. This data is needed to calculate the relative entropy.

	Entropy of a column of train_data
Pregnancies	3.167
Glucose	2.817
BloodPressure	2.362
SkinThickness	2.341
Insulin	2.922
BMI	2.754
DPF	3.155
Age	3.221
Outcome	0.9331

Table 4.1.7: In this table the Kullback-Leibler divergence of all columns can be seen across all synthetic datasets. Bigger numbers suggest bigger difference.

	gans _public	**vae _public**	**vae _private**	**gretel _public**	**gretel _corrupted**
Pregnancies	0.4388	0.7513	0.2239	0.1236	0.1184
Glucose	0.3567	0.4322	0.3723	0.1562	0.1549
BloodPressure	0.3196	0.5338	0.2109	0.0598	0.0585
SkinThickness	0.3255	0.2721	0.3658	0.0872	0.0924
Insulin	0.2955	0.6002	0.3945	0.0611	0.0559
BMI	0.2382	0.5468	0.1158	0.0963	0.1028
DPF	0.2734	0.6901	0.4010	0.2630	0.25
Age	0.4192	0.7773	0.4101	0.0859	0.0865
Outcome	0.0026	0	0	0.0026	0.0039
Mean	0.2966	0.5115	0.2771	0.1040	0.1026

distance. The maximum distance can be calculated as the dataset itself is scaled to be between 0 and 1. It can be calculated as the distance between the lowest possible values and the highest possible values.

$$\operatorname{dist}(p, q) = \sqrt{9 \cdot (1-0)^2} = 3 \tag{4.1.9}$$

The maximum distance, as calculated in Equation 4.1.9, is 3. As the minimum distance is 0, and the distance is getting bigger in a linear fashion with bigger values, we can divide all computed distances with 3 to normalize them. Now the distances can be interpreted in a new way. After normalizing, distances can be seen as the percentage to highest distance possible. For example, the distance 0.05 would mean 5% of the biggest distance possible. From this point, all shown distances will be normalized. As can be seen in Figure 4.1.1, all synthetic datasets vary at least 3% from the closest row in the training dataset. One exception is the **gretel_corrupted** dataset, in which similar rows as in the training dataset were inserted, which is exactly the purpose of the dataset. As a by-product, we have a list of distances that can be used to calculate the mean of the distance to the nearest neighbor used in subsection 4.1.4.

4.2 METRICS MEASURING THE ADVERSARY ESTIMATE

The issue with using the metrics, applied in section 4.1, is that they do not give an estimate if the data is actually private or not private. They give an estimate if the synthetic data algorithm produced data that is similar to the real data in terms of usability. In the control dataset, **gretel_corrupted**, data was inserted that is too similar to original data. This could be confirmed in subsection 4.1.6. It is to test if this really is a privacy concern. The adversary, that want to find out sensitive data from individuals, would not know which rows in the synthetic dataset are close to the original dataset. To test this, I will propose metrics that are necessary to declare the data as not private. Given the synthetic data, the adversary would estimate the values of sensitive information of the real data. The adversary is assumed to know identifying information of the individuals in the dataset as well as the values of quasi-identifiers of all individuals in the dataset.

4.2.1 *Adversary Estimate Error*

Idea: It is pretended that an adversary tries to estimate the sensitive values of individuals. We will calculate the guess of the adversary and calculate the percentage error that the adversary made with his estimate. If the percentage of the error, that the adversary makes in his estimate is too low, the privacy of the dataset needs to be considered violated.

Implementation concept: As the adversary is assumed to know all identifying information and quasi-identifiers, he will estimate the sensitive values for each individual. For each set of identifying information and quasi-identifiers in the real data, we will find k similar sets of identifying information and quasi-identifiers in the synthetic data. The k similar rows will be calculated using the k-Nearest

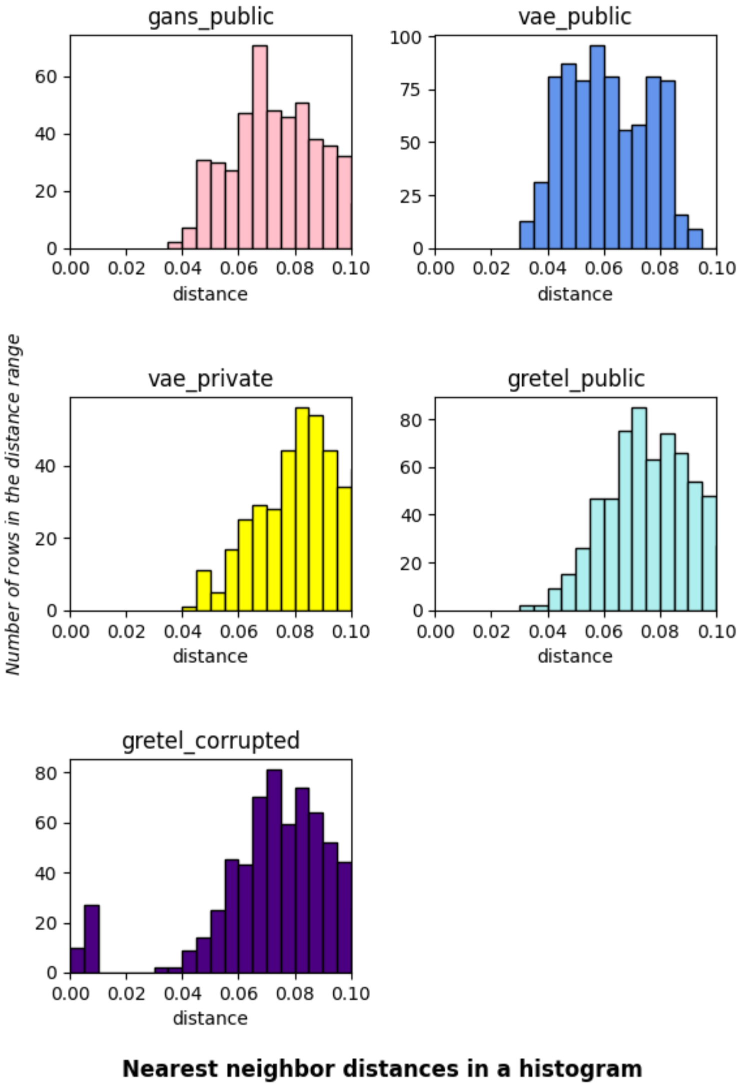

Figure 4.1.1: Histogram of distances of the each row in the synthetic dataset to the closest training dataset row.

Table 4.2.1: Error percentage for the adversary. The adversary estimates the value of the sensitive data by looking at the values of the nearest neighbor (k=1) of his known quasi-identifiers.

	gans _public	**vae _public**	**vae _private**	**gretel _public**	**gretel _corrupted**
Glucose	38.4	30.9	32.5	28.3	27.6
BloodPressure	30.6	19.9	23.9	22.6	21.9
SkinThickness	45.9	31.9	67.3	31.3	29.9
Insulin	50.9	35.3	35.9	43.9	42.2
DPF	54.7	47.2	52.0	66.9	62.9
Outcome	28.7	41.9	46.9	32.5	29.5
Mean	44.1	33.04	42.32	37.5	35.6

Neighbors method. Meaning, for each row in the training dataset, the k rows in the synthetic dataset that have the smallest euclidean distance to it, will be chosen. The estimate of the sensitive value is calculated as the arithmetic mean of the sensitive values. For all k and each kind of sensitive value, the error needs to be sufficient to say that the synthetic dataset does not violate privacy. I will assume an error higher than 20% as sufficient for privacy in this work while a scientifically grounded estimate needs to be made in the future. The percentage error [27] that the adversary makes can be summarized by relative change Equation 4.2.1.

$$\text{Error percentage} = \frac{|\text{true value} - \text{estimated value}|}{0.5 \cdot (|\text{true value}| + |\text{estimated value}|)} \cdot 100\% \qquad (4.2.1)$$

Implementation for the synthetic datasets generated in this work: In our dataset, there is no identifying information. The adversary would use a combination of quasi-identifiers instead to find out the sensitive information. These would be "Pregnancies", "Age" and "BMI". The results of the measure for $k = 1$ can be seen in Table 4.2.1. Only for the **vae_public** dataset, for the column "BloodPressure", an estimate could be made that is closer than 20% to the original value. The approach is applied to all k. For other k values, the estimate is similar to the result of $k = 1$ for all datasets. In Table 4.2.2 the estimate error percentage of the adversary can be seen for the **vae_public** with k values of $1, 3, 7, 13$ and 21.

4.2.2 *Adversary estimate of sensitive values using known data of unique individuals*

Idea: In subsection 4.2.1, the adversary made an estimate over all rows of the data. In this section, I want to test if the estimate of the adversary can be made better if the adversary only calculates estimates for rows that are more 'unique'. The basic idea behind calculating the adversary's estimate is here the same as in subsection 4.2.1. 'Unique' can mean that the individuals equivalence class is small or that the difference between an individuals identifying information and quasi-identifiers is different enough from the identifying information and

Table 4.2.2: Error percentage for the adversary. The adversary estimates the value of the sensitive data by looking at the values of the nearest k neighbors of his known quasi-identifiers, The data comes from the **vae_public** dataset.

	k=1	**k=3**	**k=7**	**k=13**	**k=21**
Glucose	30.9	30.2	29.6	29.	28.6
BloodPressure	19.9	19.9	19.9	20.0	19.9
SkinThickness	31.9	31.1	30.0	29.8	29.7
Insulin	35.3	34.6	34.3	34.1	33.9
DPF	47.2	47.0	46.8	46.8	46.7
Outcome	41.9	38.4	37.1	34.7	34.5
Mean	34.5	33.5	33.0	32.4	32.2

quasi-identifiers of other individuals. For this metric follows, as for the previous metric, if the percentage of the error, that the adversary makes in his estimate is too low, the privacy of the dataset needs to be considered violated. The estimates will be only made for the 'unique' rows of the original dataset.

Implementation concept: Calculate the rows of the original dataset that are most 'unique' in the identifying information and quasi-identifiers. The row's identifying information and quasi-identifiers are considered 'unique' when the euclidean distance to the nearest neighbors, in the same dataset, are higher compared to others. I will look at the most 'unique' 1% rows. As the adversary needs to make an estimate on the synthetic dataset, the closest set (using the euclidean distance) of identifying information and quasi-identifiers will be taken. The estimation process from subsection 4.2.1 will be then applied to these rows. For all 'unique' rows, from 1 to number of the 'unique' rows in the dataset, the estimate of the adversary needs to be lower than a threshold. As in subsection 4.2.1 I will assume the error of 20% as sufficient for privacy.

Implementation for the synthetic datasets generated in this work: Here, 1% of the rows, hence 7, most 'unique' rows in the original dataset were found, and the closest rows, in the synthetic dataset, to these, were taken. In Table 4.2.3, the error percentage of adversary's estimate can be seen for $k = 1$. The estimates are a lot closer than for the estimate using all combination of quasi-identifiers of the original dataset. The only synthetic dataset where the adversary did not make an estimate higher than 20 %, is the **vae_public** dataset. The best estimate of the adversary could be made on "BloodPressure" with only an error of 7.9%. In Table 4.2.4, the error percentage of adversary's estimate can be seen for different k for the estimated of 7 rows of the **gans_public** dataset.

Table 4.2.3: Error percentage for the adversary. The adversary took the most 'unique' sets of quasi-identifiers and took only similar synthetic rows for estimates. The adversary estimates the value of the sensitive data by looking at the values of the nearest neighbor ($k = 1$).

	gans _public	**vae _public**	**vae _private**	**gretel _public**	**gretel _corrupted**
Glucose	39.1	32.9	28.0	17.1	18.0
BloodPressure	8.7	21.5	12.7	11.3	7.9
SkinThickness	37.3	37.7	33.2	31.0	29.2
Insulin	62.8	61.9	16.7	33.3	25.9
DPF	45.8	69.0	31.9	51.9	50.9
Outcome	28.6	42.9	28.6	28.5	14.3
Mean	37.0	44.3	25.2	28.8	24.4

Table 4.2.4: Error percentage for the adversary. The adversary estimates the value of the sensitive data by looking at the values of the nearest neighbor of corresponding rows from the **gans_public** to 'unique' sets of quasi-identifiers.

	k=1	**k=3**	**k=5**	**k=7**
Glucose	39.1	33.7	33.5	28.8
BloodPressure	8.7	12.3	14.9	14.5
SkinThickness	37.3	32.2	36.1	35.2
Insulin	62.8	56.1	53.3	45.9
DPF	45.8	47.9	53.6	50.4
Outcome	28.5	42.8	42.8	42.8
Mean	37.0	37.5	39.0	36.3

5 CONCLUSION

Aside from creating a synthetic data generation algorithm, generating synthetic data with machine learning algorithms is not a trivial task. Given an implemented algorithm, without bugs, there are various parameters that need to be adjusted to produce similar data. It becomes a matter of trial and error. Over-fitting the real dataset with wrong parameters can lead to good looking data. However, if the data is created to assure privacy of individuals of the dataset, over-fitting would destroy the privacy of the individuals [5]. An adversary could, with a high precision, estimate the values of the original dataset, given that the adversary knows the identifying information and quasi-identifiers in the dataset. In [5] it is argued, using similarity metrics, the generated dataset should not be too similar as the original dataset. In many other works [10] [11], an argument about the privacy of data is not made after the creation of said data.

The goal of this thesis was to give a metric of privacy that can be applied to synthetic data that was generated. The adversary-estimate-metrics I introduced, show when the privacy is violated. They use an estimate of an assumed adversary and label the data as not private if the estimate of the adversary is high. The exact threshold, when the data should be considered not private, is still to be determined.

In section 4.1 was measured how close the generated datasets were to the real dataset. These metrics showed that **gretel_corrupted**, the dataset where similar rows of data were inserted, has the highest similarity to the real data. Second closest was the **gretel_public** dataset with around 10% difference. Other datasets would not be considered as not private, as the similarity of the synthetic data and the real data was low. In subsection 4.2.1 two things were noticed. First, the **gretel_corrupted** dataset, the dataset that has the most similarity, can be used to give an adversary a good estimate for some individuals, see Table 4.2.4. For this reason, I would confirm that similarity can give an estimate over privacy. The second thing that was shown is that some synthetic datasets, that were considered too dissimilar by the metrics in section 4.1, could also be used to give an adversary a good estimate of the sensitive information of individuals, e.g. **gretel_public** dataset gave could be used to give an adversary good estimated of the medical information about **gretel_public**, shown in Table 4.2.4.

An advantage of the introduces adversary-estimate-metrics, is also that they can be applied regardless of the size of the generated synthetic dataset as well as the size of the original dataset. For most used data similarity metrics, the sizes of datasets need to be the same. Often it is desired to create big synthetic datasets. Uses in machine learning require high amounts of data. Giving an estimate of privacy with most similarity metrics would not be possible, making the introduced adversary-estimate-metrics a good choice.

Data is collected more and more, e.g. with smart watches, smart homes and everyday life devices. The data of the individuals can provide valuable information for society, from learning if a pedestrian walkway needs to be enlarged due to high traffic to creating algorithms that can identify deceases on CT scans. Such observations are only possible if there is a lot of data, which needs to be anonymous due to legal and ethical reasons. One key role could be the generation of private synthetic data. The possibility to create private datasets from measured data can be valid for the discovery of correlations that were not seen before. The results of this thesis are one step towards measuring the privacy of synthetic data to ensure that more data is published to drive research forward.

A

APPENDIX

Table A.0: In this table an example of the first rows of the Pima Indians Diabetes dataset can be seen. The values are taken from the original dataset with no changes.

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
0	118	64	23	89	0	1.731	21	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
...	...	...	...	...	...	...	...	...

BIBLIOGRAPHY

[1] *What is personal data?* en. Text. URL: https://ec.europa.eu/info/law/law-topic/data-protection/reform/what-personal-data_en (visited on 01/14/2021).

[2] Chris Culnane, Benjamin Rubinstein, and Vanessa Teague. „HEALTH DATA IN AN OPEN WORLD.“ en. In: (), p. 23.

[3] Steven M Bellovin. „Privacy and Synthetic Datasets.“ en. In: (), p. 39.

[4] Jerome P. Reiter. „New Approaches to Data Dissemination: A Glimpse into the Future (?)“ en. In: *CHANCE* 17.3 (June 2004), pp. 11–15. ISSN: 0933-2480, 1867-2280. DOI: 10.1080/09332480.2004.10554907. URL: http://www.tandfonline.com/doi/full/10.1080/09332480.2004.10554907 (visited on 08/20/2021).

[5] Khaled El Emam, Lucy Mosquera, and Richard Hoptroff. *Practical Synthetic Data Generation*. en. O'Reilly Media. ISBN: 978-1-4920-7274-4. URL: https://learning.oreilly.com/library/view/practical-synthetic-data/9781492072737/ (visited on 01/04/2021).

[6] *Enabling developers and organizations to use differential privacy*. en. URL: https://developers.googleblog.com/2019/09/enabling-developers-and-organizations.html (visited on 08/22/2021).

[7] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. „Privacy Loss in Apple's Implementation of Differential Privacy on MacOS 10.12.“ In: *arXiv:1709.02753 [cs]* (Sept. 2017). arXiv: 1709.02753. URL: http://arxiv.org/abs/1709.02753 (visited on 08/22/2021).

[8] Franziska Boenisch. „Differential Privacy: General Survey and Analysis of Practicability in the Context of Machine Learning.“ en. In: (), p. 107.

[9] Isabel Wagner and David Eckhoff. „Technical Privacy Metrics: A Systematic Survey.“ en. In: *ACM Computing Surveys* 51.3 (July 2018), pp. 1–38. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3168389. URL: https://dl.acm.org/doi/10.1145/3168389 (visited on 12/21/2020).

[10] Vincent Bindschaedler, Reza Shokri, and Carl A. Gunter. „Plausible deniability for privacy-preserving data synthesis.“ en. In: *Proceedings of the VLDB Endowment* 10.5 (Jan. 2017), pp. 481–492. ISSN: 2150-8097. DOI: 10.14778/3055540.3055542. URL: https://dl.acm.org/doi/10.14778/3055540.3055542 (visited on 01/04/2021).

[11] Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, and Latanya Sweeney. „Privacy Preserving Synthetic Data Release Using Deep Learning.“ en. In: (), p. 17.

[12] *Pima Indians Diabetes Database*. en. URL: https://kaggle.com/uciml/pima-indians-diabetes-database (visited on 08/12/2021).

[13] *Pima Indians Diabetes Database*. en. URL: https://kaggle.com/uciml/pima-indians-diabetes-database (visited on 08/16/2021).

[14] Olga Troyanskaya, Mike Cantor, Gavin Sherlock, Trevor Hastie, Rob Tibshirani, David Botstein, and Russ Altman. „Missing Value Estimation Methods for DNA Microarrays.“ In: *Bioinformatics* 17 (July 2001), pp. 520–525. DOI: 10.1093/bioinformatics/17.6.520.

[15] *SAP-samples/security research differentially private generative models*. original-date: 2019-08-21T22:33:31Z. Dec. 2020. URL: https://github.com/SAP-samples/security-research-differentially-private-generative-models (visited on 01/04/2021).

[16] fzhurd. *A Step by Step Guide to Generate Tabular Synthetic Dataset with GANs*. en. Feb. 2021. URL: https://medium.com/analytics-vidhya/a-step-by-step-guide-to-generate-tabular-synthetic-dataset-with-gans-d55fc373c8db (visited on 08/14/2021).

[17] Sanford Weisberg. „Yeo-Johnson Power Transformations.“ en. In: (), p. 4.

[18] Chandrasekaran Anirudh Bhardwaj, Megha Mishra, and Kalyani Desikan. „Dynamic Feature Scaling for K-Nearest Neighbor Algorithm.“ en. In: (), p. 10.

[19] Rahul Bhadani. *AutoEncoder for Interpolation*. Jan. 2021.

[20] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J Radke, and Octavia Camps. „Towards Visually Explaining Variational Autoencoders.“ en. In: (), p. 10.

[21] *Welcome to Gretel!* URL: https://docs.gretel.ai/ (visited on 08/14/2021).

[22] *gretelai/gretel-synthetics*. original-date: 2020-03-02T15:54:44Z. Dec. 2020. URL: https://github.com/gretelai/gretel-synthetics (visited on 01/04/2021).

[23] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. „Generative Adversarial Networks: An Overview.“ en. In: *IEEE Signal Processing Magazine* 35.1 (Jan. 2018). arXiv: 1710.07035, pp. 53–65. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2765202. URL: http://arxiv.org/abs/1710.07035 (visited on 08/14/2021).

[24] Bauke Brenninkmeijer. „On the Generation and Evaluation of Tabular Data using GANs.“ en. In: (), p. 70.

[25] John Tabak. *Geometry: The Language of Space and Form*. en. Google-Books-ID: roHuPiexnYwC. Infobase Publishing, May 2014. ISBN: 978-0-8160-6876-0.

[26] Neeraj Kumar, Li Zhang, and Shree Nayar. „What Is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images?“ en. In: *Computer Vision – ECCV 2008*. Ed. by David Forsyth, Philip Torr, and Andrew Zisserman. Vol. 5303. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 364–378. ISBN: 978-3-540-88685-3 978-3-540-88688-4. DOI: 10.1007/978-3-540-88688-4_27. URL: http://link.springer.com/10.1007/978-3-540-88688-4_27 (visited on 08/22/2021).

[27] *Relative change and difference*. en. Page Version ID: 1027738431. June 2021. URL: https://en.wikipedia.org/w/index.php?title=Relative_change_and_difference&oldid=1027738431 (visited on 08/23/2021).