How realistic is my synthetic data? A qualitative approach

Applied Data Science Masters Thesis

Tryfon Rigas Tzikas
t.tzikas@students.uu.nl

First supervisor
Dr. A.A.A. (Hakim) Qahtan

Second supervisor
Dr. Yannis Velegrakis

Universiteit Utrecht

Department of Information and Computing Sciences
Utrecht University
Netherlands
July 2022

Abstract

Missing values represent one of the most common challenges for data analytics tasks. For that reason, a lot of techniques have been proposed to fill the missing values through what is called "Data Imputation". Recent studies on generating synthetic data demonstrate that Generative Adversarial Networks (GANs) can be used to effectively solve this problem as follows: for each example in the original data generate a synthetic example that keeps the existing values. The generated example should contain values for the features with missing values. However, to confirm if GANs can provide significant improvements over traditional data imputation techniques, we need a technique to measure the quality of the generated examples. The quality of the generated example can be measured by determining how realistic the synthetic data is compared to the original examples. In this project, we develop a tool for successfully measuring the quality of the synthetic data. We compare the quality of the generated data using GANs to other synthetic data generation techniques.

Keywords— Synthetic Data, Missing Values, Evaluation, Generative Adversarial Networks

Acknowledgements

I would like to spare this section to thank the people who supported me throughout this period. Without their help and continuous guidance, I would not have been able to work on such a challenging topic. First of all, I would like to thank my first supervisor Dr. Hakim Qahtan for his continuous guidance and attention. Our weekly meetings have been crucial for my progress and without his feedback and his domain expertise I could not have achieved such a quality result for the scientific community. I would also like to thank my second supervisor Dr. Yannis Velegrakis for his valuable feedback and his support for the project. Finally, I would like to thank my family, my dear parents, and my sister for always supporting me at every educational step of my life. Every accomplishment and success that I had so far is because of their love and affection throughout these years.

Contents

Chapter 1

Introduction

The technological advances in recent years, together with the digitalization of traditionally manual processes, have led to the extensive and continuous collection, process, and analysis of data. According to a report by IBM [1] more than 2.5 quintillion bytes of data are being created every day. This enormous amount of data has enabled evidence-based methods for decision-making instead of the traditional experienced-based decision-making. The scope of this thesis is to evaluate the utility of the data for the evidence-based methods and the consistency of newly generated data.

Statistical analysis and pattern recognition algorithms have emerged rapidly to find patterns and new insights from historical data. These methods require high-quality data for the analysis to be complete and reliable. Low data quality includes incomplete data, inconsistent data, duplicated data, and poor data security. The most common challenges are incomplete data and data security. Incomplete data occurs when specific data is not available or not stored in a given dataset and can lead to misinterpreting the data or even not being able to use specific algorithms. Reasons for data that are not available (also called missing values) can vary from user's mistakes during data collection, poor data maintenance, defective hardware, and many others. This concerns all the domains where large amounts of data are collected and later stored and analyzed. Data security refers to the challenge of data to protect sensitive and private information.

In most cases, users care about the statistical information about the data, and not the sensitive information. Therefore, a solution that optimizes the trade-off between data quality and data privacy is synthetic data: an artificial copy of the original data that carries the same statistical information. Synthetic data can tackle the challenge of incomplete data by taking the corresponding synthetic value and imputing the missing value, balancing the dataset by creating samples in the minority groups, and introducing a first layer of data protection since the synthetic data do not exist in real life.

Various synthetic data generators have been developed in the last years because of their efficiency and their ability to offer solutions to a variety of challenges. Nevertheless, empirical evidence of their utility needs to be further explored. In order for synthetic data to be valuable, it is essential that they are evaluated in terms of utility; their ability to capture the statistical information, and security; their ability to protect the original data.

1.1 Motivation

The purpose of this thesis is to create a framework that evaluates synthetic data. The newly generated data need to be evaluated regarding their ability to copy the statistical features and to offer privacy guarantees without making assumptions about the original distribution. Synthetic data with high scores from the framework contains "realistic" data. This evaluation framework can be used by synthetic data generators to measure their performance and also by data analysts to decide whether they can use the synthetic data in a meaningful way.

There are several cases where synthetic data can be extremely useful and therefore it is crucial to measure their utility. First, missing values can be imputed with values that derive from newly generated synthetic samples. The imputed values need to be evaluated on how much they "agree" with the rest of the original data. Secondly, synthetic data can balance datasets that consist of classes with fewer examples than others. These synthetic samples need to have the same distribution as the rest of the minority group.

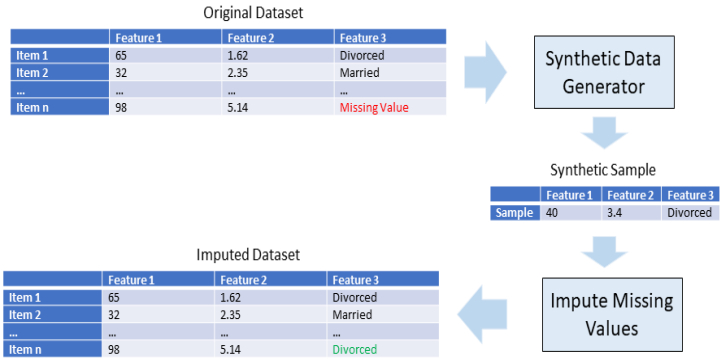


Figure 1.1: Missing values imputation using synthetic data

1.2 Research Questions

The main research of this thesis is to develop a framework that evaluates synthetic data and their utility. Depending on the task of the synthetic data, the framework needs to determine whether the imputed missing values are realistic and typical considering the rest of the original data. Thus the main research questions are the following:

Research Question 1: How can we develop a framework to evaluate synthetic data?

Research Question 2: How can the implemented framework take into account both categorical and numerical variables?

Research Question 3: How can we evaluate each synthetic sample individually?

1.3 Outline

The thesis consists of 5 chapters, which are the introduction, related work, theoretical background, proposed framework, experimental evaluation, and conclusion. After the introduction, related work includes detailed research on the literature review and relevant work that has been conducted until now on the subject. The theoretical analysis chapter introduces the algorithms and the theory that the reader of the thesis needs to be familiar with. In

the proposed framework chapter the structure of the framework and the methodology that was followed are explained in detail. The experimental evaluation chapter shows the experiments that were conducted and the evaluation of the framework. The last chapter includes conclusions of the thesis and future work to be done.

Chapter 2

Related Work

Humanity is collecting an enormous amount of data - and very often there are missing values. Missing values are data that are not stored or present for specific variables in a given dataset and is the most common problem across different disciplines, such as medical research, government agencies, and private companies. With growing interest in data-driven tools like machine learning, data quality becomes essential for the results of the analysis. Therefore, missing values adversely impact the data quality and the accuracy of the final outcome [2]. Reasons for missing values in a dataset may vary: data corruption due to improper maintenance, intentional error from the user at the time of data entry, hardware errors, and so on.

Missing values are categorized into three main groups. Missing Completely At Random (MCAR), where there is the same probability of data being missed for all the observations. In this case, there is no correlation between the missing value and any other observed or unobserved values. Missing At Random (MAR), where the missing values depend on other observed variables; and Missing Not at Random (MNAR), where missing values depend on unobserved data [3].

2.1 Techniques for Handling Missing Values

Many approaches from statistical models to deep learning algorithms have been developed over the years to handle missing values. The most popular techniques are deletion, imputation, maximum likelihood techniques, and the creation of synthetic data.

Deletion: Deleting all rows that consist of missing values is the easiest and simplest approach. Deletion can be 'complete deletion' or 'list-wise deletion', where all rows which have missing values are deleted, 'specific deletion' in which only rows that have more missing values than a predefined percentage are deleted, and 'pair-wise deletion' where only rows that have missing values in the variables that are used for the analysis are deleted [4].

Imputation: Imputation techniques replace missing values with newly created ones, based on the existing information of the dataset. The new values can be mean, mode, median, predicted by a classifier or by using K Nearest Neighbors (KNN) [5]. In mean and mode imputation techniques, values are replaced by the mean or mode of the non-missing values of the same attribute. In techniques involving a classifier, a model is trained on the existing values of an attribute and then it predicts new values to replace the missing data. Methods based on KNN algorithms, use the k nearest samples to impute missing data [6].

Synthetic Data: Recently synthetic data has attracted attention because of the variety of challenges that aims to solve. Data privacy, unbalanced datasets, and missing values are only a few of the areas in that synthetic data generators can provide significant results. We next provide a brief description of the main approaches for creating synthetic data: Bayesian Networks, Categorical latent Gaussian process, Synthetic Minority Over-Sampling Technique, and Generative Adversarial Networks.

Bayesian Networks (BN) are probabilistic graphical models where each variable is represented as an edge and the nodes represent the dependencies between the variables. For synthetic data, the graph represents the real data and the relationships among them and offers a visual representation of them. To generate new data we sample from the inferred Bayesian network. BN can provide useful insights about relationships among the variables and in a computationally efficient way [7].

Categorical Latent Gaussian Process(CLGP) is a model for generating multivariate categorical data [8]. By using a non-linear transformation of a continuous latent space it produces vectors of categorical variables. This approach combines the linear categorical Gaussian model, the Gaussian process latent variable model, and the Gaussian process classification. This approach has a rich latent non-linear mapping that can capture complex distributions but scales poorly with data size.

Synthetic Minority Over-Sampling Technique (SMOTE) is a way to create synthetic data by duplicating samples in the minority class[9]. SMOTE selects samples that are close in the feature map, draws a line between them in the feature space, and creates a new point along that line.

Generative Adversarial Networks (GANs) have spurred the discussion about unsupervised machine learning tasks and specifically for creating synthetic data [10]. GANs are essentially two different neural networks that are trained jointly in a competitive manner. The first one tries to create realistic synthetic data, whereas the second one tries to distinguish between real and synthetic data. Each network pushes the other to perform better. Although GANs initially could not handle categorical variables many extensions have been suggested to deal with mixed data types. GANs provide significant results in synthetic data generation but training them and tuning the hyper-parameters remain difficult tasks.

2.2 Evaluation of Synthetic Data

A wide variety of synthetic data generators have been developed in recent years to meet the ever-growing demands for inclusive data sharing, data privacy, and handling missing values in large datasets. However, more research needs to be conducted in order to evaluate their utility. In this section, we present the main suggestions and advances in evaluation metrics in synthetic data.

Utility measures have been categorized into two main groups: narrow metrics and broad metrics [11]. Narrow metrics evaluate the performance of synthetic data on a specific task on the original data, whereas broad measures capture general characteristics of the entire dataset (differences in marginal distribution, similarity between two datasets). Snoke et al. [12] had categorized measures by the same philosophy referring to them as specific and general measures accordingly.

Dankar et al.[13] compare different synthetic data generators by categorizing further the evaluation metrics and defining multiple dimensions of utility (quality dimensions). They classify existing utility metrics based on the measure they attempt to preserve: attribute fidelity, bivariate fidelity, population fidelity, and application fidelity. Attribute fidelity refers to the measures that evaluate the basic structural similarity between the data. Synthetic data must

have the same structure and aggregated statistics (variable types, formats, names, means, ranges) or similar univariate distributions for continuous and discrete variables. The most popular techniques to measure attribute fidelity are Hellinger distance [14] and Kullback-Leibler divergence [15]. Bivariate fidelity covers statistical dependencies between the variables. It is measured by calculating pairwise correlation using heatmaps [16] or by calculating pairwise correlation difference [7]. Population fidelity measures the correlation between the entire distribution. Most popular techniques are cross-classification metric measures [7], log-cluster metric measures [17] and distinguishability type metrics by using propensity score [12],[18],[19]. Application fidelity evaluates the performance of synthetic data on prediction tasks [20]. A machine learning model is trained on synthetic data and real data and then is tested on the real data to see how it will behave in real life.

Drechler et al. [11] propose the evaluation of data generators based on machine learning models by their ability to preserve analytical validity. Goncalves et al. [7] evaluate three types of synthetic data generators (probabilistic models, classification-based imputation models, and generative adversarial neural networks) by dividing the metrics into two groups: data utility and information disclosure. Data utility refers to the ability of synthetic data to capture the statistical features of the real data. The metrics used by this group are Kullback-Leibler divergence, pairwise correlation difference, log-cluster metric, support coverage metric, and cross-classification metric. Information disclosure measures how much real data may be retrieved from the synthetic data. Data utility may be reduced due to an increased need for data privacy that leads to generalization and smoothing of the synthetic data [21]. Therefore, both groups of measures need to be taken into consideration.

Alaa et al. [22] underline the importance of fidelity, diversity, and generalization performance of any generative model and create 3 new dimensions of evaluation metrics: α-precision, β-recall, and authenticity. They introduce a new approach to evaluating synthetic data generators where instead of looking at the entire distribution, they evaluate each sample individually as high or low quality.

- Fidelity: high fidelity means realistic samples
- Diversity: the ability of the model to produce diverse samples
- Generalization: highly generative models avoid over-fitting

Mannino and Abouzied [23] create a tool named Synner, which visualizes the characteristics of the dataset, such as each field's statistical distribution, its domain, and its relationship to other fields. Therefore, it generates real-looking data and gives instant feedback on every user interaction. Arnold and Neunhoeffer [24] evaluate data quality along two dimensions. The first dimension is when synthetic data are evaluated on training data or on an underlying population. In the second one, data quality depends on the general similarity of distributions or on performance for specific tasks (e.g. inference or prediction).

Hittmeir et al [25] empirically assess the quality of synthetic data by testing them on specific supervised machine learning tasks on publicly available datasets. For each attribute they plot the histogram showing the distribution of both the real and the synthetic data, they calculate correlation and dependencies between attributes, and finally, they calculate the distances between real and synthetic data.

Finally, Emam [26] summarizes 7 ways that the quality of synthetic data has been evaluated so far. First, they mention *replication of studies*, where analysis is performed on the real data and then replicated on the synthetic data. High quality would mean that the same conclusions are drawn from both analyses. *Subjective assessment by domain experts*, where experts can evaluate the distance between real and synthetic data. *General utility metrics*, which is the calculation of the correlation between the variables of synthetic and real data, or the creation

of a classifier to distinguish them. *Bias and stability assessment*, is the computation of general utility metrics on the real dataset, and then the calculation of the variation of these metrics in different synthetic datasets. *Structural similarity*, where synthetic data should have the same variable types and formats, variable names, metadata, and file formats as the real data. *Comparison with public aggregate data*, where statistics from synthetic data are compared with publicly available results to see if they agree. And finally, *comparison with other privacy-enhancing technologies*, where assessments can be performed on other methods of synthetic data generation, and then the results are compared.

In this thesis, a new approach is suggested where mixed-type variables are evaluated and instead of assessing the whole distribution, we evaluate each sample separately. More specifically the contribution of this work is the following:

1. We introduce two dimensions of data evaluation based on how realistic the samples are (nonoutliers) and how typical they are (whether a targeted feature is correctly generated).
2. We apply the method to both categorical and numerical variables. Gower [27] distance was selected for outlier detection, and decision trees for the supervised part that can perform both regression and classification tasks.
3. Instead of looking at all the synthetic samples collectively, we evaluate each sample individually.

Chapter 3

Theoretical Background

In this chapter, the theoretical background is analyzed. First, we define the problem statement, then we analyze the challenge of missing values and the techniques for handling them, and finally, we present Deep Learning and Generative Adversarial Networks.

3.1 Problem Statement

In this thesis, we propose a framework to evaluate synthetic data that are generated to impute missing values in the original dataset. Given a dataset $D = \{X, S, Y\}$, X is the set of attributes that do not contain any missing values, S is the set of attributes that contain missing values, and $Y \in \{0, 1\}$ is the original class label, which indicates the decision outcome. Let $|D|$ represent the cardinality (number of instances) of the dataset D. We assume without loss of generality, that D is an imbalanced dataset with missing values where, for example, $G,\ G' \in S_i$ where G represents the existing values of the instances G' represent the missing values. Let $D_{Complete}$ be the subset of D, which excludes all the instances that have at least one missing value and $D_{Incomplete}$ is the subset of D which includes instances with missing values. A synthetic data generator can be trained on $D_{Complete}$, and given as conditions the existing values X it can generate samples that will impute the missing values G' with the generated ones $\hat{G}$. The generated values $\hat{G} \in S_i$ need to be evaluated for whether they carry the statistical information of the original data.

We denote the real and the generated data as $X_r \sim P_r$ and $X_g \sim P_g$, respectively, where $X_r, X_g \in X$, with P_r and P_g being the real and generative distributions, and X being the input space. The real and synthetic datasets are $D_{real} = \{X_{r,i}\}_{i=1}^{n}$ and $D_{synth} = \{X_{g,i}\}_{j=1}^{m}$, where each sample follows the real and generative distribution respectively. Our goal is to construct a metric $\mathcal{E}\{D_{real}, D_{synth}\}$ that measures the quality of D_{synth} in order to evaluate the performance of the synthetic data generator and audit the model outputs by discarding (individual) "low quality" samples. Our evaluation method should be able to tell if any given (individual) sample $X_g \sim P_g$ is of high or low quality.

During this research the following problems will be tackled:

- Evaluating synthetic samples based on the original distribution.
- Identifying limitations in existing evaluation techniques.
- Developing a framework that contributes to the limitations of the existing approaches.

In Chapter 4, we propose a two-dimensional evaluation metric:

$\mathcal{E} = (Regularity, Fidelity)$

Regularity expresses how realistic a synthetic sample is, and fidelity measures the probability that a synthetic sample resides in the real distribution. Let $S_r = supp(P_r)$ be the support of the real distribution and $S_g = supp(P_g)$ be the support of the generative distribution. The distribution P is divided into "normal" samples concentrated in S, and "outliers" residing in $\bar{S}$, where $S = S \cup \bar{S}$ [22].

3.2 Missing Values

Data Science involves many underlying fields such as Statistics, Mathematics, and Programming. The main purpose is to extract knowledge and insights from noisy, structured, or unstructured data by applying scientific methods. Data need to be in a certain form and fulfill specific requirements regarding data quality. Data quality is mostly achieved through the handling of missing values.

Data Science starts from cleaning, aggregating, and reformatting the data so that they are ready for specific ways of processing. The analysis is the stage when data scientists develop algorithms, analytics, and AI models and provide patterns and predictions for business decision-making. These insights need to be evaluated through scientifically designed experiments.

- Capture: collecting structured or unstructured data from any available source, either from sensors, web scraping, or manual entry.
- Prepare and maintain: transforming the data into a consistent format. This includes using ETL (extract, transform, load) technologies to upload the data to a data warehouse or a data lake.
- Preprocess or process: find biases, statistical features, ranges, and distributions to determine which methods apply to the data.
- Analyze: apply analytical methods to extract insights, patterns, and predictions.
- Communicate: visualize the results and present the information that was extracted by using specific tools.

It is clear that when there is incomplete data (instances with missing values) the analysis can be negatively influenced. Statistical methods and machine learning algorithms rely greatly on the consistency of the data. Therefore, missing values can lead to low-quality results, misinterpretation of the patterns within the data, and limitations on the algorithms that can be used for the analysis.

3.3 Statistical Methods for Imputation

Statistical methods for data imputation use the features and characteristics of the existing data. One of the most common approaches is through KNN (K-Nearest Neighbors). Configuration of the KNN involves selecting the distance metric (e.g. Euclidean) and the number of contributing neighbors (the k hyperparameter of the algorithm). A new sample is generated by finding the samples in the real distribution that are closest to it and averaging these nearby points to fill in the value.

Imputation Using Multivariate Imputation by Chained Equation (MICE) is a method where missing values are filled in multiple times. Multiple imputations (MIs) are more efficient than

a single imputation as they measure the uncertainty of the missing values. Finally, stochastic regression imputation tries to predict missing values by regressing them from other related variables in the same dataset plus some random residual value.

3.4 Deep Learning

In the last years, a technique that has been proven highly efficient is generating synthetic data with Deep Learning and then imputing missing values with the new samples. Deep learning is a sub-field of machine learning that is based on artificial neural networks. The term "deep" refers to the use of multiple layers in the network. Since Deep Learning is far more complicated than the previous techniques, it is worth analyzing how the algorithms work and how we can generate new data with synthetic data generators.

3.4.1 Deep Neural Networks

As mentioned before artificial neural networks consist of a set of nodes (artificial neurons) that imitate the functions of the human brain. Nodes are connected through synapses that transmit a signal to other nodes. Nodes process the signals that are transmitted to them and generate an output signal through non-linear functions (activation functions). Nodes and synapses have a weight, which is adjusted during the learning process. The activation function $f : R^M \longrightarrow R$ is defined as:

$$f(\vec{x}) = act\left(\sum_{i=1}^{M}(x_i w_i)\right) + b = \vec{w}^T\vec{x} + b$$

where w_i are the synaptic weights, x_i the output signals, and b the bias.

Typically neurons are aggregated into layers. The signals travel from the first layer (input layer) to the last layer (output layer). The output of the network is:

$$\vec{y} = act(W\vec{x} + \vec{b})$$

where W is the MxN array of weights, and b is the bias.

In order for the network to be optimized and learn from the training data, an optimization algorithm must be chosen. Backpropagation is the algorithm that is most commonly used in the training process of artificial neural networks. The method computes the gradient of the loss function concerning the weights of the network for a single input-output example.

The loss function computes the error between the predicted and the actual values of the output layer of the network. The most commonly used loss functions are:

1. Mean Squared Error:

   $J_{MSE} = \frac{1}{M}\sum_{i=1}^{M}(y - \widehat{y_i})^2$

   where $y_{i,c}$ are the network's predictions
2. Cross Entropy Loss:

   $J_{CrossEntropy} = -\sum_i y_{i,c} log(p_{i,c})$

where $y_{i,c}$ is the binary variable of whether class c is the correct prediction for the observation i and $p_{i,c}$ is the probability that i belongs at class c.

The error of the network between the desired output $p_{i,c}$ and the predicted output $p_{i,c}$ is calculated through the loss function C:

$E = C(y, \widehat{y})$

Next, for every node j with an activation function ϕ and net_j the weighted sum of the previous outputs, we define the output o_j :

$o_j = \varphi\left(net_j\right) = \varphi(\sum_{k=1}^{n} w_{kj} o_k)$

The derivative of the error of the loss function is calculated by applying the chain rule twice:

$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial w_{ij}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}$

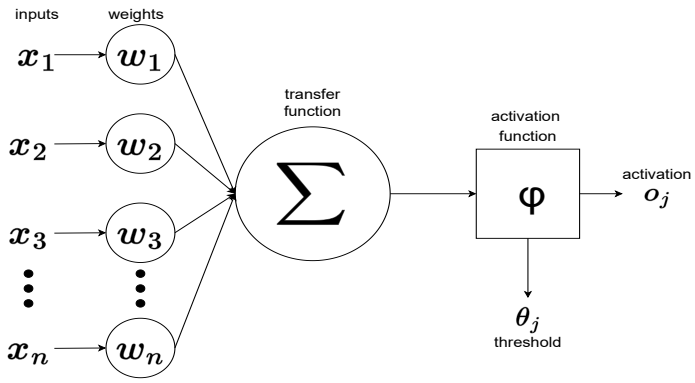


Figure 3.1: Calculation of a neural network output

3.4.2 Generative Adversarial Network

Generative Adversarial Networks (GANs) are an approach of generative modeling using deep artificial neural networks. The specific architecture was designed by Ian Goodfellow et al. [10] in June 2014.

Generative modeling is an unsupervised learning task where the algorithm needs to discover and learn the regularities and patterns of the training data and automatically generate new samples.

GANs are an architecture that transforms the task into a supervised problem by creating two sub-models that compete with each other: the generator and the discriminator. The generator (generative network) generates new candidates while the discriminator (discriminative network) tries to classify the examples as either real (from the training data) or fake (generated). The two sub-models are trained at the same time in a zero-sum game, adversarial, until the generator can fool the discriminator by generating realistic data.

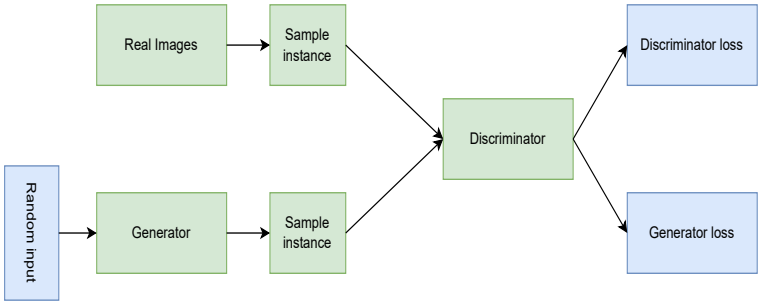


Figure 3.2: Basic GAN architecture

Both of the sub-models are neural networks. The discriminator (D) is connected directly to the output of the generator (G). Through backpropagation, the discriminator's classifications inform the generator on how to update its weights. D is trained to maximize the probability of correctly classifying training examples and samples from G. G on the other hand, is trained to minimize the same probability. The function that the two sub-models play in the minimax game is the following:

$$E_x[log(D(x))] + Ez[log(1 - D(G(z)))]$$

In this function:

- E_x is the expected value over all real data instances.
- D(x) is the discriminator's estimate that a real instance x is real.
- G(z) is the generator's output when given noise z.
- D(G(z)) is the discriminator's estimate that a fake instance is real.
- E_z is the expected value over all random inputs to the generator.
- The formula derives from the cross-entropy between the real and generated distributions.

GANs have been increasingly used in many applications due to their high performance in generating realistic examples. The main domains that GANs are used for are image creation and audio synthesis.

Chapter 4

Proposed Framework

The methodology that was followed in this thesis, is divided into two parts (a) Outlier Detection, (b) Feature Classification. Firstly, by introducing Gower distance we calculate the distance between all the mixed type variables, and then by applying a clustering algorithm (DBSCAN) we determine whether a sample is considered an outlier. This represents how "realistic" a synthetic sample is. Secondly, we train a decision tree algorithm (TensorFlow Decision Trees - TFDF) on the original data, and we evaluate the generated sample by predicting the targeted column. This represents how "typical" the synthetic sample is and whether it agrees with the statistical distribution of the original dataset.

To evaluate the synthetic data we introduce a two-dimension metric to quantify whether a sample is considered an outlier and whether the targeted feature is correctly generated by the SGD, thus answering the first research question in Chapter 1. Our method is applied at sample-level evaluation instead of assessing a collection of synthetic data.

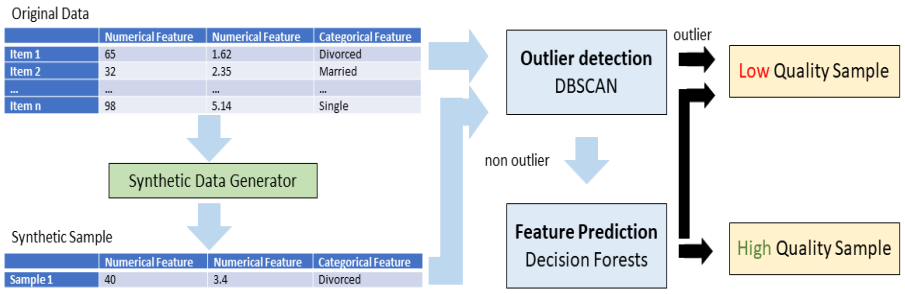


Figure 4.1: Implemented framework

4.1 Methodology

The detailed methodology that was followed can be seen in the flowchart below. Each step of the process aims to the evaluation of the synthetic sample as of high or low quality.

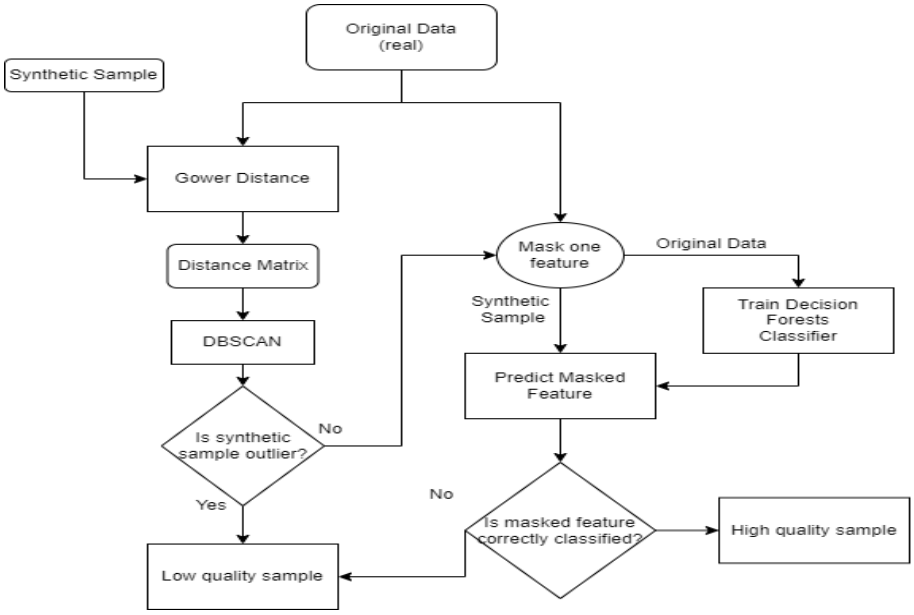


Figure 4.2: Methodology that as followed

4.2 Preprocessing

As mentioned before, the implemented framework of this thesis considers both numerical and categorical variables. Gower distance is used to measure how different two records are, and therefore to offer a solution to the second research question in Chapter 1.1. The general form of the coefficient is the following:

$D_{Gower}(x_1, x_2) \ = \ 1 - (\frac{1}{p} \sum_{j=1}^{p} s_j(x_1, x_2))$

For each feature $k \ = \ 1, 2, .., p$ we define a score $s_{ijk} \in [0, 1]$. If x_i and x_j are close to each other along feature k, then s_{ijk} is close to 1. Conversely, if they are far apart along feature k, the score is close to 0.

The way that the score s_{ijk} is computed is determined by the type of the feature k. A quantity δ_{ijk} is also defined: if x_i and x_j can be compared along feature k, then $\delta_{ijk} = \ 1$. If x_i and x_j cannot be compared along feature k (e.g. because of missing values), then $\delta_{ijk} = \ 0$. Gower's distance follows the formula below:

$s_{ijk} = \ \frac{\sum_{k=1}^{p} s_{ijk} \delta_{ijk}}{\sum_{k=1}^{p} \delta_{ijk}}$

The score s_{ijk} can be computed for either quantitative variables (categorical variables) or qualitative variables (categorical variables):

1. Quantitative variables: $s_{ijk} = \ 1 \ - \ |x_{ik} \ - \ x_{jk}| \ / \ R_k$, where R_k is the range of feature k
2. Qualitative variables: $s_{ijk} = \ 1, x_{ik} = x_{jk}$

If s_{ijk} defines how similar x_i and x_j are, then $\sqrt{1 \ - \ s_{ij}}$ is the distance between them. If there are no missing values, the distance satisfies the triangle inequality and can therefore be considered as valid distance. For any 3 observations x_i, x_j and x_l :

$$\sqrt{1 - s_{ij}} + \sqrt{1 - s_{jl}} \geq \sqrt{1 - s_{il}}$$

After having calculated the Gower distance between every sample and for every feature, we get the distance matrix (two-dimensional array) that contains the distances, taken pairwise, between the elements of the set. This allows us to use specific clustering algorithms for outlier detection.

$$Distance\ Matrix = \begin{bmatrix} 0 & \cdots & d_{ij} \\ \vdots & \ddots & \vdots \\ d_{ji} & \cdots & 0 \end{bmatrix}$$

where d_{ij} is the distance between x_i and x_j.

4.3 Main Components

4.3.1 Outlier Detection

It is crucial to determine whether the generated sample can exist in real life. This is done through anomaly detection and more specifically outlier detection algorithms. Clustering algorithms that can perform with distance matrices as input, are k-medoids, hierarchical clustering algorithms, and DBSCAN. DBSCAN has been proven to detect more efficient outliers ([28], [29] , [30]). DBSCAN algorithm views clusters as areas of high density, separated by areas of low density. Due to this rather generic view, clusters found by DBSCAN can be any shape, and therefore the algorithm performs well for outlier detection. The core characteristic of DBCAN is the core samples that are at the center of high-density areas. Therefore, a cluster consists of a set of core samples, each close to each other (measured by some distance measure) and a set of non-core samples that are close to a core (but not core samples themselves). The two main parameters of the algorithm are minSamples and epsilon. MinSamples is the number of samples in a neighborhood for a point to be considered as a core sample and epsilon is the maximum distance for two points to be considered in the neighborhood of the other.

The figure below shows how DBSCAN creates clusters with minSamples = 3. A circle with radius epsilon is drawn around every data point. All data points with at least 3 neighbors are considered core samples and represented by green. Samples that have less than 3 neighbors but greater than 1 are considered border samples and are represented by yellow. Finally, samples that have no neighbors are considered noise (outliers) and are represented by red.

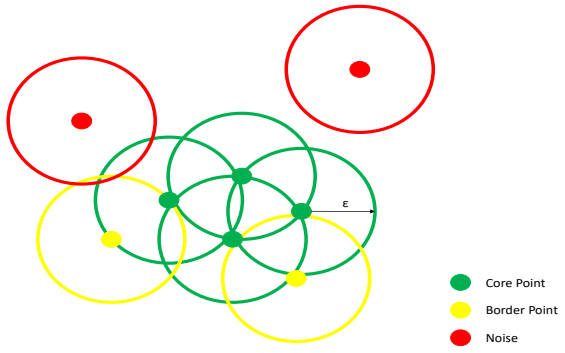


Figure 4.3: Clusters created by DBSCAN with minSamples = 3

4.3.2 Feature Prediction

After having determined whether the generated sample is considered a non-outlier and therefore is realistic, we have to investigate further whether the sample is typical. The newly generated samples have to be typical in terms of following the statistical characteristics and the distribution of the original dataset. In order to do that we use decision trees. Decision trees are a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It consists of a hierarchical tree structure with root, node, branches, internal nodes, and leaf nodes.

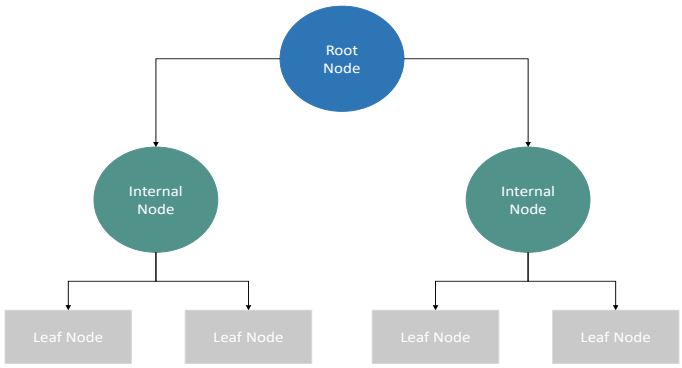


Figure 4.4: Decision tree

As it is shown above, a decision starts from the root node, which connects with the outgoing branches to the internal nodes (or decision nodes). Based on the available features, both root and internal nodes divide the data to form homogenous subsets, which are denoted by leaf nodes. Leaf nodes show all possible outcomes of the dataset.

In this way, we can predict the class or value of a target variable by learning decision rules inferred from training data. To evaluate synthetic data with decision trees, we define a target variable of the original data to train the decision tree. Afterward, we mask the target variable of the generated synthetic sample, and we predict it using the rest of the features. If the prediction of the decision tree and the actual value of the synthetic sample agree, then we evaluate the sample as high quality, otherwise as low quality.

Decision trees' high accuracy and flexibility allow us to evaluate the synthetic sample by predicting any feature of the sample. This can be particularly useful when we are generating new samples to fill in missing values. Therefore, we are only interested in one feature and whether the generated value of this feature "makes sense" considering the statistical distribution of the original data. We use as a target variable the one which contains missing values, and check whether the prediction and the value of the generated sample agree. In addition, decision trees are robust in predicting the target value for minority groups. Finally, the algorithm's ability to perform both classification and regression is an excellent way to handle mixed-type variables.

In order to obtain the highest accuracy when predicting any feature, TensorFlow Decision Forests (TFDF) were used. TFDF is a collection of state-of-the-art algorithms for classification and regression that was introduced by TensorFlow in August 2021. It includes

algorithms such as random forests, gradient boosted trees, CART, (Lambda)MART, DART, Extra Trees, greedy global growth, oblique trees, one-side-sampling, categorical-set learning, random categorical learning, out-of-bag evaluation, and feature importance, and structural feature importance. All models are trained simultaneously and they vote for the class with the highest probability. The final prediction is based on the summary of the votes.

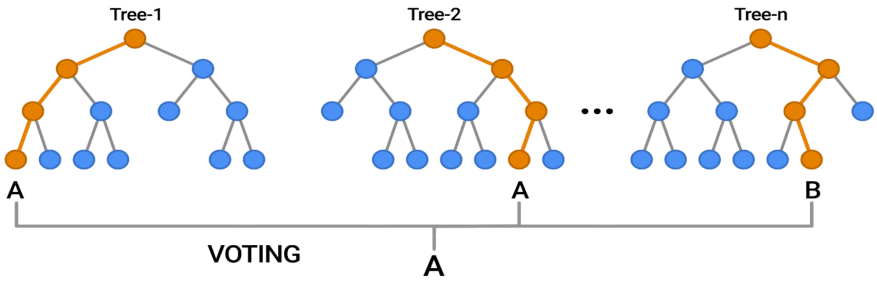


Figure 4.5: TensorFlow Decision Forests

4.4 Evaluation

The final part of our framework is to evaluate the synthetic samples individually as of low or high quality. In this part, we answer the third research question in Chapter 1.1. If a generated sample is not identified as an outlier by DBSCAN and the target variable agrees with the prediction of TFDF, then we evaluate the sample as of high quality. On the other hand, if the sample is identified as an outlier or if the target variable does not agree with the prediction, then the sample is considered of low quality.

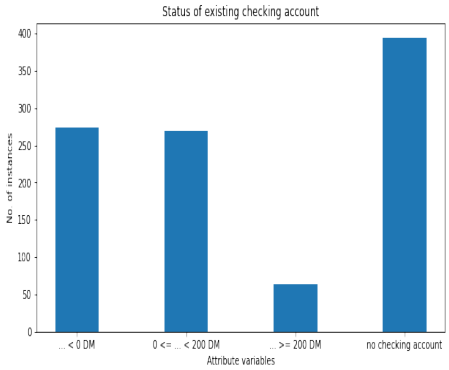
Chapter 5

Experimental Evaluation

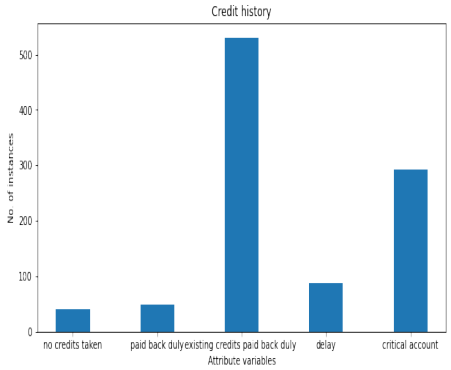
This chapter demonstrates the accuracy of the algorithms that were used and the overall performance of the implemented framework. First, we perform experiments for analyzing the outlier detection algorithm, and then we test the TensorFlow decision forests. Since our goal is to impute missing values with generated samples, TensorFlow decision forests need to predict accurately multiple features (that are most likely minority classes).

5.1 Datasets

The experiments were conducted mainly on two datasets: the swiss banknote and the german credit dataset. The swiss banknote dataset consists of 200 banknotes in total, 100 genuine, and 100 counterfeits. It is suited for testing outlier detection algorithms. The German credit dataset consists of 1000 instances, 20 attributes, and 1 binary decision label. There are 7 numerical, and 13 categorical attributes and the dataset is imbalanced. These characteristics provide us an excellent opportunity to perform experiments and optimize our classifier to predict both categorical and numerical variables and use as labels different attributes with imbalanced data.



(a) Imbalanced



(b) Imbalanced

5.2 Metrics

In this section, we explain the metrics that were used to evaluate our algorithms and the overall performance of the framework.

1. Confusion Matrix: a matrix that visualizes and summarizes the performance of a classification algorithm for each feature. Each row of the matrix represents the real samples, and each column represents the predicted samples.

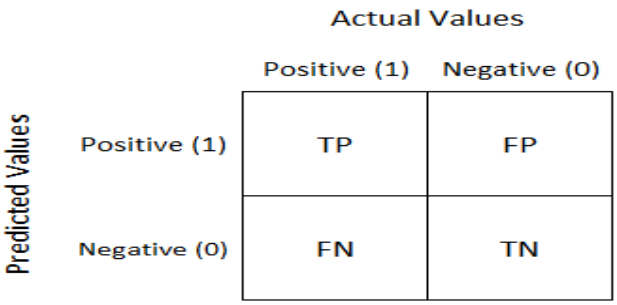


Figure 5.2: Confusion matrix

where TP (True Positives): when the algorithm predicted positively and the output was positive. When there are multiple classes, this is the case where the class that was predicted was the same as the real one.

TN (True Negatives): when the algorithm predicted negative and the output was negative.

FP (False Positive): when the algorithm predicted positive but the output was negative.

FN (False Negative): when the algorithm predicted negative but the output was positive.

2. $Accuracy = \frac{TruePositives + TrueNegatives}{TotalNumberofSamples}$
3. $Precision = \frac{TruePositives}{TruePositives + FalsePositives}$
4. $Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$
5. $F1Score = \frac{Precision \times Recall}{Precision+Recall}$
6. Mean Square Error (MSE): for numerical variables, we use the metric MSE
   $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)$
   where Y are the observed values, $\hat{Y}$ are the predicted values, and n are the total samples.

5.3 Outlier Detection Results

DBSCAN was fine-tuned and tested on the swiss dataset, in order to detect outliers (counterfeit) banknotes. Samples that are annotated at cluster -1, are considered to be outliers by the algorithm. As we can see from the classification report below, the algorithm's performance was significantly high.

Classification Report			
	Precision	Recall	f1-score
genuine	0.98	0.92	0.95
counterfeit	0.92	0.98	0.95
accuracy			0.95

5.4 Classification Results Using Different Target Variables

In order to see the model's behavior, we trained it multiple times, and each time we used a different column as a target variable. TFDF algorithm trains 4 decision forests (Random Forest Model, Gradient Boosted Trees Model, CartModel, Distributed Gradient Boosted Trees Model), and each one of them votes for the class with the highest probability. After summarizing all the votes, we choose the optimal prediction. To compare our approach we use the CartModel alone, as a baseline. Below we can see the accuracy for the classification results (the highest the better), and on the second diagram, we can see the MSE for the regression results (the lowest the better).

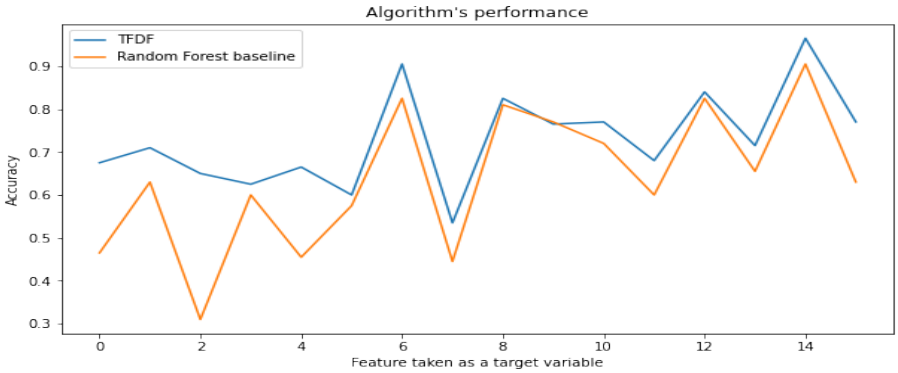


Figure 5.3: TFDF accuracy on categorical target variables

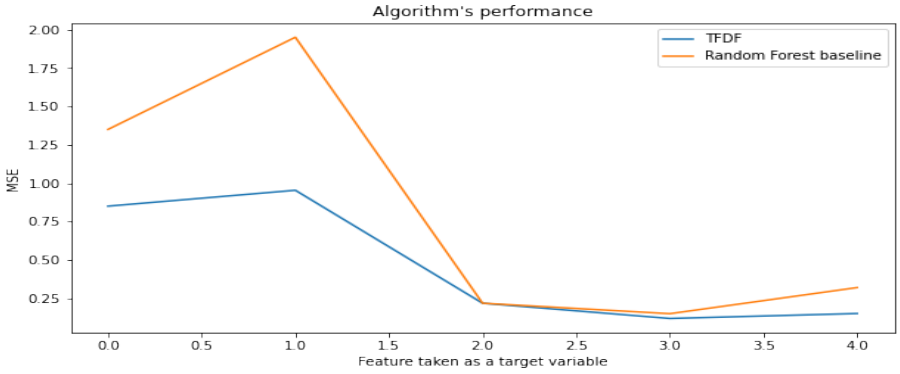
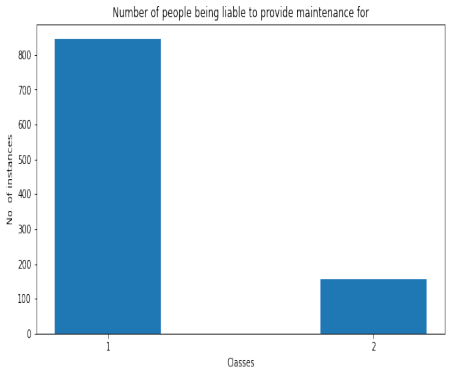


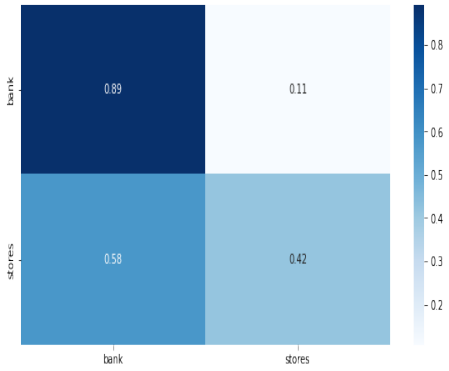
Figure 5.4: TFDF MSE on numerical target variables

As we can see above, TFDF in general does not drop below 70% accuracy and outperforms

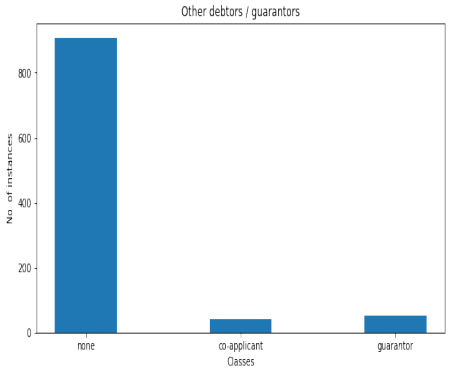
almost with any given variable the baseline model. In addition, as shown in the figures below, the algorithm performs well even for the minority classes where our dataset is highly imbalanced. The confusion matrix is normalized over the true (rows) so that its performance is more clearly illustrated.



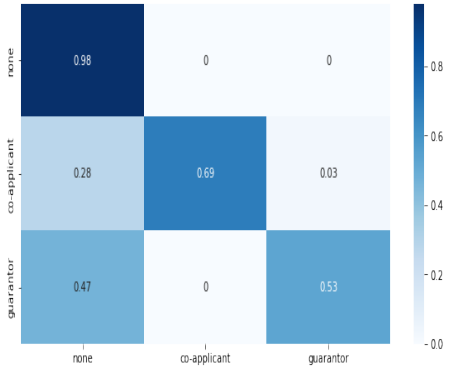
(a) Class Distribution



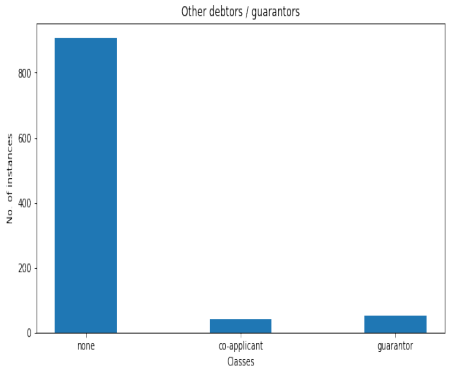
(b) Confusion matrix



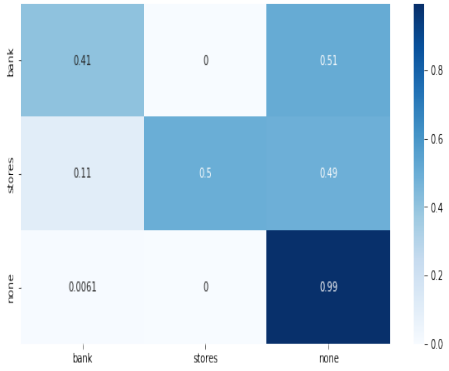
(a) Class Distribution



(b) Confusion matrix



(a) Class Distribution



(b) Confusion matrix

5.5 Framework Performance

Finally, for the overall evaluation of the framework, 100 new samples were generated with CTGAN (Conditional GAN). For each sample, all the values of the attributes were given as a condition, and only the final label was generated with CTGAN. This allowed us to annotate the samples with "0" if the generated label was different from the original sample, and "1" if the generated label was the same. Afterward, we evaluated the samples as high or low quality using the framework, and we test whether the framework's score and the annotated labels agree. The accuracy of the framework was 78.55%. Below we can see the confusion matrix for each class.

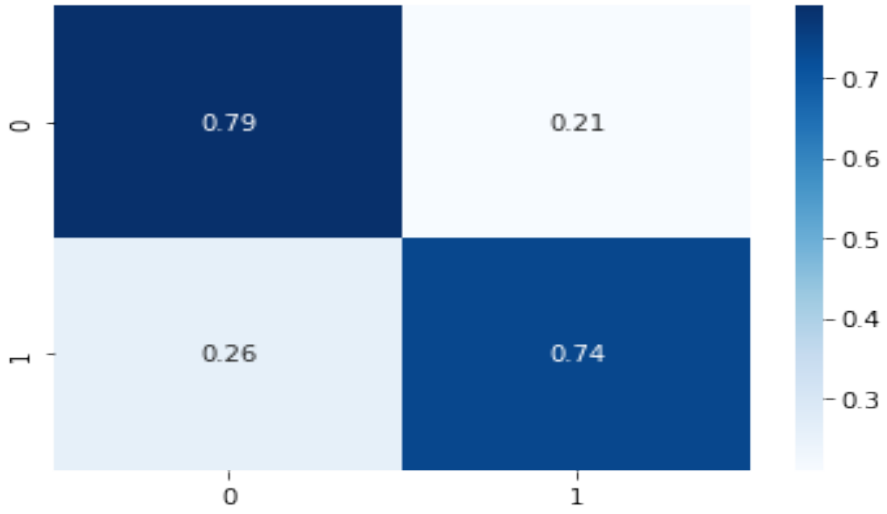


Figure 5.8: Framework performance evaluation

Chapter 6

Conclusion

In this chapter, the conclusions from the experiments and the proposed framework are presented.

6.1 Summary

In this thesis, the topic of synthetic data evaluation is researched. Evaluating synthetic data is a difficult and subjective task. Nevertheless, it is crucial to calculate the utility of the newly generated data so that we can determine whether they can replace the original data or whether they can be used for further analysis. The ultimate goal of the synthetic data that is examined in this thesis is imputing missing values with values that derive from synthetic samples. Most of the existing approaches do not consider datasets with mixed-type variables and emphasize only on numerical values. Furthermore, most techniques evaluate the synthetic distribution by looking at all the synthetic samples.

The proposed framework of this thesis offers a two-dimensional evaluation of mixed-type variables and assesses each sample individually as being of high or low quality. The proposed framework does not assume the type of the variables of the dataset and can evaluate a synthetic sample based on every feature. This can be of great significance since it allows us to emphasize on the feature that contains missing values and that we ultimately want to impute. It consists of two main components: outlier detection and feature classification.

First, Gower distance is calculated from synthetic data so that we can later perform techniques for mixed-type variables. The distance matrix is a matrix that consists of the distance of each element pair-wise. We use the distance matrix input to the DBSCAN, and we detect outliers so that we can determine whether a generated sample is considered an outlier regarding the original dataset. Afterward, we train a classifier multiple times, using as target variable all the features of our dataset. Therefore, we can predict any column of synthetic samples and evaluate whether they agree with the prediction. So overall, the proposed framework achieves a two-dimensional score of the quality of the synthetic sample by emphasizing a specific feature.

Multiple experiments are conducted to determine the optimal techniques for each component of the framework and evaluate the overall performance. Several outlier detection techniques were tested and DBSCAN was selected based on their performance and the literature. Experiments on the Gower distance were also performed to fine-tune the weights of each variable and not give a higher weight to categorical variables. Classifiers for mixed-type variables were

tested and an ensemble method was chosen combining many decision tree algorithms using the TensorFlow TFDF library. Finally, new samples are generated to evaluate the framework's performance. The proposed framework is flexible and robust in evaluating synthetic data.

6.2 Answers to Research Questions

The answers to the research questions in this thesis are found in Chapter 4 and can be confirmed by the results in Chapter 5. In order to answer the first and third research questions, this thesis followed the methodology in [22] where a three-dimensional evaluation metric is proposed. Similarly, we propose a two-dimensional evaluation metric where we evaluate the regularity of a synthetic sample (non-outlier) and its fidelity (the probability that the sample resides in the real distribution). The framework classifies each sample individually as being of high or low quality. Finally, we answer the second question in Chapter 4 were to take into account both categorical and numerical variables we use Gower distance. Most of the existing approaches in the literature analyze numerical variables without giving an efficient solution to mixed-type variables.

6.3 Limitations

There are several limitations identified to the proposed framework. The first limitation concerns datasets with significant-high dimensionality which can affect the Gower distance. The more dimensions are inserted in the formula, the lower weight is given to each feature, and therefore the harder it gets to identify outliers that exceed the range of only a few variables. Another limitation of the framework is related to the choice of outlier detection as the main component. If the goal of the synthetic data is to create outliers, then evaluating them as low-quality samples will not be useful for our main goal. Finally, the classification algorithms that were used, need to be fine-tuned for every different dataset to obtain the highest performance. This means that if we want to predict every feature of a given dataset (and thus evaluate a synthetic sample) we need analyze the dataset and properly fine-tune the decision trees that are used.

6.4 Future Work

There are several directions to be investigated further to improve the proposed framework. The first direction would be to research more synthetic data generators for the final evaluation of the framework. Generating more realistic synthetic samples could improve the analysis and evaluation of the proposed framework. Another direction would be to evaluate separately categorical and numerical data so that evaluate the framework's performance for each variable type and apply changes accordingly. Finally, another future direction could be to evaluate more classifiers to predict different features. Choosing the right algorithm can affect crucially the utility of the proposed framework. The features of a given dataset could be divided into categories based on statistical characteristics (numerical, categorical, range, standard deviation) and the optimal algorithm for each category will be chosen.

Bibliography

[1] *10 Key Marketing Trends for 2017 and Ideas for Exceeding Customer Expectations*. Bizibl.Com. Accessed: Sep. 1, 2020. DOI: https://bizibl.com/marketing/download/10-key-marketing-trends-2017-and-ideas-exceeding-customer-expectations.

[2] Tsai Chih-Fong and Chang Fu-Yu. "Combining instance selection for better missing value imputation". In: *Journal of Systems and Software* 122 (Dec. 2016). DOI: 10.1016/j.jss.2016.08.093.

[3] Judi Scheffer. "Dealing with Missing Data". In: *Research Letters in the Information and Mathematical Sciences* 3 (June 2002).

[4] P. Jonsson and C. Wohlin. "An evaluation of k-nearest neighbour imputation using Likert data". In: *10th International Symposium on Software Metrics, 2004. Proceedings.* 2004, pages 108–118. DOI: 10.1109/METRIC.2004.1357895.

[5] Ms. R. Malarvizhi and Dr. Antony selvadoss Thanamani. "05-07 5 K-Nearest Neighbor in Missing Data Imputation". In: 2012.

[6] Madan Yadav and Basav Roychoudhury. "Handling Missing Values: A study of Popular Imputation Packages in R". In: *Knowledge-Based Systems* 160 (Nov. 2018), pages 104–118. DOI: 10.1016/j.knosys.2018.06.012.

[7] André Gonçalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Sales. "Generation and evaluation of synthetic patient data". In: *BMC Medical Research Methodology* 20 (May 2020). DOI: 10.1186/s12874-020-00977-1.

[8] Yarin Gal, Yutian Chen, and Zoubin Ghahramani. "Latent Gaussian Processes for Distribution Estimation of Multivariate Categorical Data". In: (Mar. 2015).

[9] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *J. Artif. Intell. Res. (JAIR)* 16 (June 2002), pages 321–357. DOI: 10.1613/jair.953.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. "Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems* 3 (June 2014). DOI: 10.1145/3422622.

[11] Jörg Drechsler and Jerome Reiter. "An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets". In: *Computational Statistics Data Analysis* 55 (Dec. 2011), pages 3232–3243. DOI: 10.1016/j.csda.2011.06.006.

[12] Joshua Snoke, Gillian Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. "General and specific utility measures for synthetic data". In: *Journal of the*

Royal Statistical Society: Series A (Statistics in Society) 181 (Apr. 2016). DOI: 10.1111/rssa.12358.

[13] Fida Dankar, Mahmoud Ibrahim, and Leila Ismail. "A Multi-Dimensional Evaluation of Synthetic Data Generators". In: *IEEE Access* 10 (Jan. 2022), pages 1–1. DOI: 10.1109/ACCESS.2022.3144765.

[14] Lucien Cam and Grace Yang. *Asymptotics in Statistics : Some Basic Concepts*. Jan. 2002. DOI: 10.1007/978-1-4684-0377-0.

[15] Ashish Dandekar, Remmy A. M. Zen, and Stéphane Bressan. "Comparative Evaluation of Synthetic Data Generation Methods". In: 2017.

[16] Beata Nowok. "Utility of synthetic microdata generated using tree-based methods". In: 2015.

[17] Mi-Ja Woo, Jerome Reiter, Anna Oganian, and Alan Karr. "Global Measures of Data Utility for Microdata Masked for Disclosure Limitation". In: *Journal of Privacy and Confidentiality* 1 (Apr. 2009). DOI: 10.29012/jpc.v1i1.568.

[18] Gillian Raab, Beata Nowok, and Chris Dibben. "Guidelines for Producing Useful Synthetic Data". In: (Dec. 2017).

[19] David Hand. "Synthetic Datasets for Statistical Disclosure Control: Theory and Implementation by Jörg Drechsler". In: *International Statistical Review* 80 (Dec. 2012). DOI: 10.1111/j.1751-5823.2012.00196_15.x.

[20] Debbie Rankin, Michaela Black, Raymond Bond, Jonathan Wallace, Maurice Mulvenna, and Gorka Epelde. "Reliability of Supervised Machine Learning Using Synthetic Data in Healthcare: A Model to Preserve Privacy for Data Sharing (Preprint)". In: *JMIR Medical Informatics* 8 (Mar. 2020). DOI: 10.2196/18910.

[21] Kingsley Purdam and Mark Elliot. "A Case Study of the Impact of Statistical Disclosure Control on Data Quality in the Individual UK Samples of Anonymised Records". In: *Environment and Planning A* 39 (May 2007), pages 1101–1118. DOI: 10.1068/a38335.

[22] Ahmed Alaa, Boris van Breugel, Evgeny Saveliev, and Mihaela Schaar. "How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Models". In: (Feb. 2021).

[23] Miro Mannino and Azza Abouzied. "Is this Real?: Generating Synthetic Data that Looks Real". In: Oct. 2019, pages 549–561. ISBN: 978-1-4503-6816-2. DOI: 10.1145/3332165.3347866.

[24] Christian Arnold and Marcel Neunhoeffer. "Really Useful Synthetic Data – A Framework to Evaluate the Quality of Differentially Private Synthetic Data". In: (Apr. 2020).

[25] Markus Hittmeir, Andreas Ekelhart, and Rudolf Mayer. "On the Utility of Synthetic Data: An Empirical Evaluation on Machine Learning Tasks". In: Aug. 2019, pages 1–6. ISBN: 978-1-4503-7164-3. DOI: 10.1145/3339252.3339281.

[26] Khaled Emam. "Seven Ways to Evaluate the Utility of Synthetic Data". In: *IEEE Security Privacy* 18 (July 2020), pages 56–59. DOI: 10.1109/MSEC.2020.2992821.

[27] Gulanbaier Tuerhong and Seoung Bum Kim. "Gower distance-based multivariate control charts for a mixture of continuous and categorical variables". In: *Expert Systems with Applications* 41.4, Part 2 (2014), pages 1701–1707. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.08.068. URL: https://www.sciencedirect.com/science/article/pii/S0957417413006891.

[28] Zohreh Akbari and Rainer Unland. "Automated Determination of the Input Parameter of DBSCAN Based on Outlier Detection". In: *Artificial Intelligence*

Applications and Innovations. Edited by Lazaros Iliadis and Ilias Maglogiannis. Cham: Springer International Publishing, 2016, pages 280–291.

[29] Mete Çelik, Filiz Dadaşer-Çelik, and Ahmet Şakir Dokuz. "Anomaly detection in temperature data using DBSCAN algorithm". In: *2011 International Symposium on Innovations in Intelligent Systems and Applications*. 2011, pages 91–95. DOI: 10.1109/INISTA.2011.5946052.

[30] Tran Manh Thang and Juntae Kim. "The Anomaly Detection by Using DBSCAN Clustering with Multiple Parameters". In: *2011 International Conference on Information Science and Applications*. 2011, pages 1–5. DOI: 10.1109/ICISA.2011.5772437.