# Variational Autoencoders For Tabular Data Generation (SynthVAE)

*David Brind*

NHSX

Analytics Unit - Innovation Branch

Internship Final Report

June 2022
Under the supervision of Jonathan Pearson

## Abstract

We investigated the utility of variational autoencoders (VAEs) for single table data generation. This report follows on from prior work performed by Dominic Danks. In this report we introduce a new release of SynthVAE that leverages new pre-processing techniques as well as a deeper tool library. Our findings indicate that SynthVAE provides good single table generation under the distributional metrics we measure. However, more work is needed to translate this statistical measure into downstream task performance. A wider discussion around privacy and fairness is needed to ensure utility of the model in all potential NHS use cases.

# Contents

# 1 Executive Summary

This report details the continuation of developing a variational autoencoder (VAE) with differential privacy for the purpose of demonstrating the feasibility of this algorithm to generate synthetic data for healthcare with intrinsically defined privacy. This project continues upon the previous work of building the autoencoder by considering a Gaussian Mixture Model (GMM) in the data pre-processing step to support non-gaussian variables; developing several improvements to the code base; and considering how to develop the model to include fairness in the synthetic data.

The project outcome was a second release of SynthVAE to Github (`https://github.com/nhsx/SynthVAE/tree/v0.2.0`) with this associated report. A persisting bug in the codebase exists at the time of publication which needs to be resolved to increase the fidelity of the model with the gaussian mixture pre-processing. Further time is also required to develop the fairness aspect into a future release.

# 2 Introduction

Machine learning research has led to promising innovation within healthcare. However, there is still a significant under utilisation of AI within the NHS. This is due to multiple factors, one of which being the sensitivity of healthcare datasets and hence, the stringent access regulations required for data access. A desire to embed more AI technology within the NHS is being driven forward through the newly released AI roadmap [1].

A research avenue which seeks to address this is through the generation of synthetic data. Specifically, we are interested in generating high fidelity, privacy preserving and fair synthetic data. Our initial investigations are centred on single table data, mimicking those you would expect from EHR data - MIMIC-III being a good example [2][3]. A significant portion of the research into this domain has been performed using generative models, specifically focusing on generative adversarial networks (GANs) - i.e. [4]. This is highlighted in a paper by Bourou et al. [5] who review all the GAN methods for tabular synthesis in finance.

The focus in this report is improving fidelity of SynthVAE as well as bringing in fairness constraints. These constraints will allow us to ”de-bias” datasets resulting in an improvement in fairness on future downstream tasks - especially important in healthcare settings where socioeconomic disparities are common. This is a problem that has generally been under-represented in research thus far, however influential events (such as COVID-19 socioeconomic disparities [6]) have resulted in an increased interest in fairness.

# 3 Related Work

See related work chapter of Danks' report [7] to look at comparative synthetic data models such as CTGAN [8].

# 4 Background

See background chapter of Danks' report [7] for information on VAEs as well as differential privacy.

## Embedding Fairness Into Synthetic Data Generation

Specific to our report is the inclusion of fairness concerns in synthetic data generation. In order to address this we will first outline the current key metrics within fairness, before then moving onto bias mitigation techniques within deep learning. These fundamentally revolve around adapting model loss functions to include fairness constraints, or using causal modelling to remove bias influencing links.

If we are interested in downstream tasks from synthetic data, we can measure the fairness of our set through metrics on the prediction variable. These can be split up into a few key metric groups:

- Group
- Subgroup
- Individual

Each of these metric groups then have specific metrics which can be applied. These are shown below in Table 1 from a review done by Mehrabi et al. [9]:

| Metric Type | | | |
|---|---|---|---|
| Name | Group | Subgroup | Individual |
| Demographic parity | ✓ | | |
| Conditional statistical parity | ✓ | | |
| Equalized odds | ✓ | | |
| Equal opportunity | ✓ | | |
| Treatment equality | ✓ | | |
| Test fairness | ✓ | | |
| Subgroup fairness | | ✓ | |
| Fairness through unawareness | | | ✓ |
| Fairness through awareness | | | ✓ |
| Counterfactual fairness | | | ✓ |

Table 1: Table 1 from review by Mehrabi et al. [9] showing the breakdown of fairness metrics and segregating them into similar groups

There are many metrics to choose from to measure fairness and to make the situation more complex, not all metrics are compatible with one another, i.e. you can usually only satisfy one metric at a time. This means that the level of fairness required is project specific and has to be re-evaluated depending on the research needs.

In order to implement fairness-constrained synthetic data generation, we have three main options:

- Post-generation processing to sample a ”fair” distribution
- Embedding fairness into the loss of the deep learning method generating data
- Causal modelling methods to remove harmful causal links

Option 1 has some obvious downsides and could even lead to a decrease in fairness. Namely, if bias is inherent the dataset, these unfair correlations are learnt by the generating model. Simply re-sampling to oversample minority groups would result in more sample points however, these unfair correlations would still be present in the set. In order to ensure true fairness, the learning process of the algorithm has to be constrained in such a way that it becomes a priority.

Option 2 is an active research area and also seen to be used in a number of commercial synthetic data offerings. In essence the idea here is that we have a loss function that not only contains your usual reconstruction loss (learning distributions of the variables in the dataset) but also a constraining term that uses a selected fairness metric as a penalty. Papers by Xu et al. [10] and Amirarsalan et al. [11] both look into the usage of this method with GANs. The general loss will be given by:

$$loss = -(ML) - \lambda(F) \tag{1}$$

Whereby $ML$ and $F$ give the original model loss, whether this be the ELBO term for VAEs or the min-max objective of GANs, and the fairness constraint imposed. We can therefore adapt the constraining effect of the fairness penalty by adapting $\lambda$. This approach has been shown to be effective in the papers quoted above, and has gathered significant interest. The main drawback is that fairness constraint has to be implemented at training time and so for different use cases you could end up re-training your model to fit the needs. To our knowledge there are very few papers using this same approach for variational autoencoders. However it is non-trivial to imagine the cross-over to VAEs whereby, an additional fairness penalty is included within the VAE loss.

This leads to option 3, causal modelling. The advantage of this method is that the fairness constraints are leveraged post-training and so only one training run has to be performed for the model. Changing the fairness constraint is easy at it is applied at generation time. Causal modelling differs from standard machine learning. In causal modelling we are interested in causal relationships between variables, where as in machine learning we are leveraging correlations in the variables to make a prediction. Correlated variables do not always display a causal relationship.

The first step in creating causal models is causal discovery. During this process we are trying to iteratively learn the causal links between variables. This can be done using a number of different algorithms [12] or by specifying these links if they are known prior. The second step is then to model the causal mechanisms. This can be done by modelling each structural equation as a separate generative model. Features are then generated in the topological order of the underlying directed acyclic graph (DAG). This can then be trained using an objective loss minimising the difference between synthetic and original distributions. Finally, in order to "de-bias" the synthetic data, we perform a do-operation i.e. an intervention on the conditional distribution. This is simply the same as removing an edge from the underlying DAG that corresponds to an unfair linkage. An example of a DAG is given below in Figure 1.

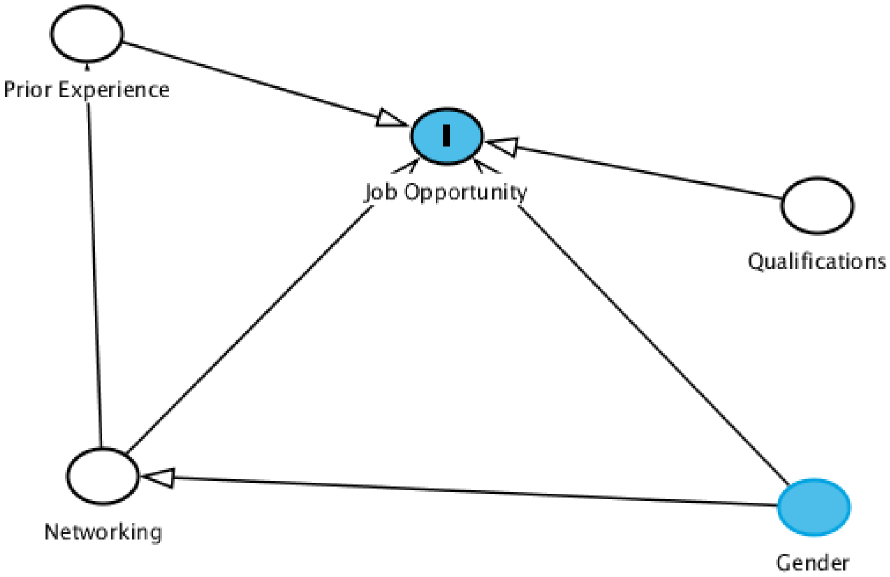


Figure 1: Figure showing the DAG representing the causal relationships between a selection of variables and job opportunity. Not made to truly reflect the causal relationships, purely for example purposes. Created using dagitty.

Van der Schaar et al. [13] show how to implement causal modelling for "de-biasing" datasets using their DECAF model. They use the PC algorithm to identify the underlying DAG structure. They then train a GAN-based model with each structural equation being represented by a separate generator. Distributions get generated in the topological order of the DAG and the discriminator then trains using the standard min-max loss differentiating between synthetic and original samples. Figure 2 taken from their respective paper [13] outlines how the process of removing edges from the DAG can satisfy certain fairness constraints. The example they give relates to demographic parity. If we want to evaluate job opportunity but we know that there is a bias towards male genders, we can perform a do operation at inference time such that we set the gender for every person as male. This removes the edge linking gender to job opportunity and we then create a fair (with respect to demographic parity on gender) dataset. Using the DAG above in Figure 1, this would remove the link between gender and job opportunity as it leads to unfair synthetic data. The only drawback to this method is that you still require background knowledge of the bias within the dataset in order to accurately perform this do-operation. Thus, no fairness related techniques can be deployed blindly, and a true understanding of the bias within your dataset is always a prerequisite.

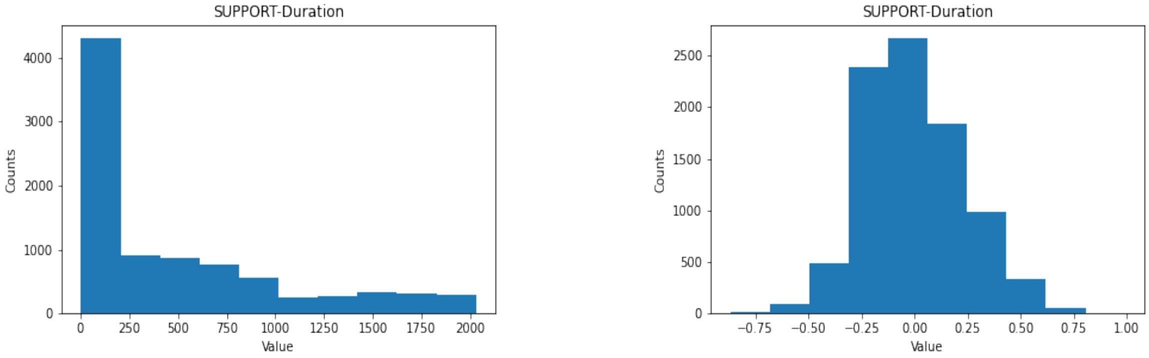
## 5 Methodology Improvements

As this project is a continuation of prior work, all of the information in chapters: Implementation and Experiments, supplied in the report by Danks [7] remains relevant. We will only present methodology and tools added to this original work.

The key difference between the prior report and this one is that we implement our methods on both the SUPPORT dataset [14] as well as a pre-processed single table extracted from MIMIC-III [2][3]. Both of these datasets provide idealistic single tables in which there is no missingness (i.e. there are no missing values present in the dataset). SynthVAE has not been extended to deal with missingness thus far. SUPPORT consists of only continuous and categorical variables, whereas, MIMIC-III single table also contains datetime variables. We one-hot encode categorical variables and we convert datetimes to continuous variables before normalisation. We rely mostly on SDV's RDT library [15] for these transformations.

## Pre-Processing for Non-Gaussian Continuous Variables

In tabular data, it is common for continuous variables to follow non-gaussian distributions. In the first iteration of SynthVAE we performed a standard scaling to normalise these variables. However, as Equation 5 from Danks' report [7] shows, we use gaussian log-likelihood as our reconstruction loss for continuous variables. This implicitly assumes that the variables follow a gaussian distribution. As a result, we need a normalisation method that both scales and transforms non-gaussian continuous variables into gaussian distributions. An example of such process is shown below in Figure 2. This uses our pre-processing method to convert the variable representing Duration within the SUPPORT dataset.

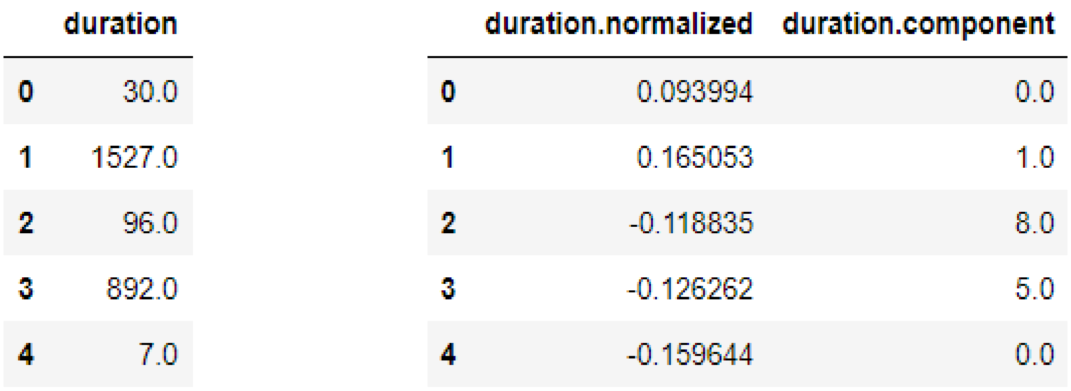


(a) Raw Duration

(b) Pre-Processed Duration

Figure 2: Figure showing the result of transforming non-gaussian continuous variables into gaussian continuous variables. This was done through performing pre-processing using variational gaussian mixture modelling.

In order to do this we leverage the GMMTransformer from the RDT library cited above. This uses the scikit-learn variational gaussian mixture modelling. Each continuous variable is fitted using this modelling approach. This separates the distribution into a mixture of differing gaussian distributions. This converts the continuous column into two columns, one with the original continuous value and the second with a categorical variable relating to the gaussian mixture it belongs to. We then normalise each mixture distribution using standard scaling and apply one-hot encoding (OHE) to the mixture category. An example of this is shown below in Figure 3 and Figure 4.



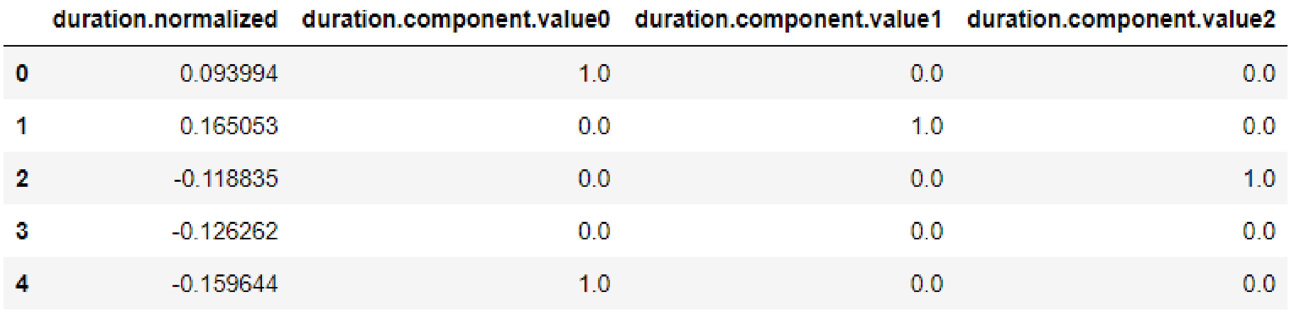
| | duration |
|---|---|
| 0 | 30.0 |
| 1 | 1527.0 |
| 2 | 96.0 |
| 3 | 892.0 |
| 4 | 7.0 |

(a) Raw Duration

| | duration.normalized | duration.component |
|---|---|---|
| 0 | 0.093994 | 0.0 |
| 1 | 0.165053 | 1.0 |
| 2 | -0.118835 | 8.0 |
| 3 | -0.126262 | 5.0 |
| 4 | -0.159644 | 0.0 |

(b) GMM Applied Duration

Figure 3: Figure showing the result of transforming the duration variable from SUPPORT using variational gaussian mixture modelling.



| | duration.normalized | duration.component.value0 | duration.component.value1 | duration.component.value2 |
|---|---|---|---|---|
| 0 | 0.093994 | 1.0 | 0.0 | 0.0 |
| 1 | 0.165053 | 0.0 | 1.0 | 0.0 |
| 2 | -0.118835 | 0.0 | 0.0 | 1.0 |
| 3 | -0.126262 | 0.0 | 0.0 | 0.0 |
| 4 | -0.159644 | 1.0 | 0.0 | 0.0 |

Figure 4: Figure showing the final form of the variable duration from SUPPORT after complete pre-processing (i.e. GMM + OHE). Note here we cut the column number to only include up to component 2 but depending on number of mixtures this could be larger.

This method does have a drawback in that, we need a sufficient number of mixtures to accurately model the continuous variable (through our work we find this to be around 8 mixtures). Depending on this value, you will then be adding up to the number of mixtures in categorical columns. This will quickly expand the dataset size and have a dominance in categorical columns.

## Hyperparameter Tuning and Early Stopping For SynthVAE

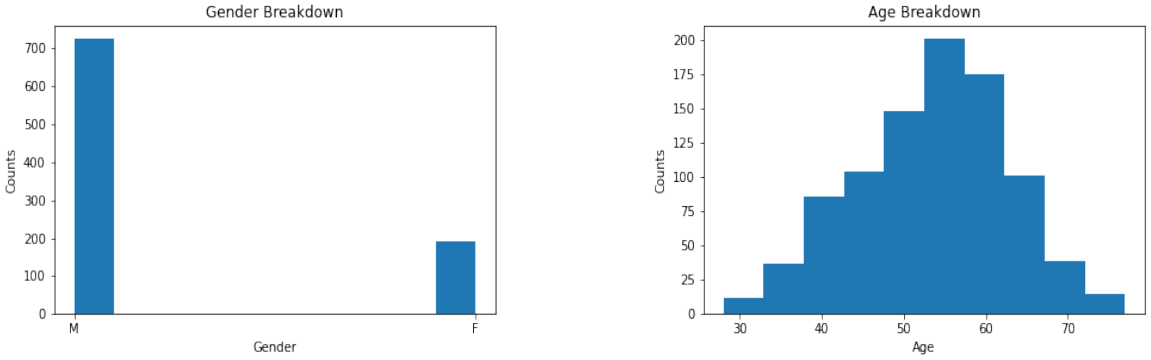
To build the codebase for SynthVAE we added in hyperparameter tuning as well as early stopping. This allows us to quickly filter out unsuccessful runs as well as find the optimal hyperparameters for each dataset. Accompanying this is the utilities to plot loss functions after training to visualise model performance. We perform hyperparameter tuning leveraging the Optuna library [16].

## Building an Example Use Case

In order to enhance the current metric toolkit, we are interested in looking at the data quality for downstream tasks. We are then also interested in measuring the fairness of our synthetic data generation. Statistical measures comparing the synthetic and the original dataset are useful, but actual utility of the data is a crucial component. To build this in we need to produce an example use case of how SynthVAE could be deployed in practise.

For this we leverage an open source dataset from kaggle based around heart failure prediction [17]. This problem is a binary classification task in which we are trying to predict if a patient has heart failure based on a number of different variables. This is still an unrealistic scenario in which we have no missing data and the dataset only contains continuous and numerical variables.

This dataset contains 918 patients and there is a large imbalance between genders, as well as an uneven distribution of age. This is shown below in Figure 5.



(a) Gender Imbalance

(b) Age Distribution

Figure 5: Figure showing the breakdown of protected variables within the kaggle heart failure dataset [17]

It is also important to understand how these protected variables differ with relation to the label of interest, heart failure, as well as with each other. The result of this aggregation is shown in Appendix A. We can see that there is a wide imbalance between the protected variables (both individually and together) and heart disease. This is useful as it gives us a good example use case to test future fairness based work.

As the actual performance of the ML model in our use case is not a priority, we favour a significantly more interpretable model. This so we can ensure that trends in the original dataset are also being picked up within the synthetic set (unless of course they cause issues with fairness of the model). For this reason we perform the classification task using decision trees. These are interpretable and allow us to view tree diagrams explaining the reasoning for classification. We perform a stratified 10 fold validation - to ensure that the same ratio of targets is in all the splits - which is repeated five times to ensure robust results.

# 6 Results

## Results on SUPPORT & MIMIC-III

For SUPPORT and MIMIC-III we perform brief hyperparameter tuning to try and find the best hyperparameters before quoting results. For MIMIC-III we introduce some constraint based sampling for certain variables. For example, for the age variable if we sample we can get negative ages that do not fit the original dataset. As a result we do post-processing sampling to ensure that all ages are greater than 0. We apply similar constraints to admission time being before first chart time and discharge time. We find that adding in these constraints helps improve the metrics of the model and this is most likely due to the variables fitting the original distributions more accurately. One issue with the metrics is that they do not scale well with size of the dataset.

We quote results for the continuous and discrete KL-divergence measures below for our MIMIC-III set. The reason for this is because it is 10-fold larger than the SUPPORT dataset and as such, calculating all the metrics is extremely time consuming. For SUPPORT runs we used a latent dimension size of 108 as well as a hidden dimension size of 640. For MIMIC-III runs we used a latent dimension size of 104 as well as a hidden dimension size of 896. It should be noted that these are quite large relative to the input dimensions of the underlying datasets, so would need further investigation into the relationship between reducing these to force a bottleneck, and the loss in fidelity of the outputs.

| SUPPORT Results | | | | | | |
|---|---|---|---|---|---|---|
| SVC Detection | GM Log Likelihood | CS Test | KS Test | KS Test Extended | Continuous KL Divergence | Discrete KL Divergence |
| 0.394433 | -34.31659 | 0.996059 | 0.9581 | 0.974375 | 0.968062 | 0.980672 |

Table 2: Table 2 shows the metrics on the SUPPORT set using the optimal model configuration found through hyperparameter tuning

| MIMIC-III Results Without Constraints | |
|---|---|
| Continuous KL Divergence | Discrete KL Divergence |
| 0.715898 | 0.844587 |

Table 3: Table 3 shows the metrics on the MIMIC-III set without constraint based sampling as well as using the optimal model configuration found through hyperparameter tuning

| MIMIC-III Results With Constraints | |
|---|---|
| Continuous KL Divergence | Discrete KL Divergence |
| 0.717175 | 0.846907 |

Table 4: Table 4 shows the metrics on the MIMIC-III set with constraint based sampling as well as using the optimal model configuration found through hyperparameter tuning

## Preliminary Results On Example Use Case

For the original training set (and all subsequent models) we use a decision tree with max depth of 3, splitting based on the gini criterion, minimum of two samples to split a node. We choose the weighted F1 score as our metric of choice to avoid any dataset imbalance issues. Over the five 10-fold validations we get the following results:

| Fold Results | | |
|---|---|---|
| Fold Number | Mean Test Score | Test Standard Deviation |
| Fold 1 | 0.8230 | 0.0478 |
| Fold 2 | 0.8372 | 0.0388 |
| Fold 3 | 0.8246 | 0.0425 |
| Fold 4 | 0.8376 | 0.0381 |
| Fold 5 | 0.8322 | 0.0418 |
| Average | 0.8309 | 0.0418 |

Table 5: Table 5 showing results of five repeated stratified 10-fold validation runs. Testing performed on original heart failure dataset

We apply the same structure for the synthetic sets however we produce five synthetic sets all with the same size of the original dataset. We average over these five sets to get a final metric. We show the average over five repeated 10-folds for each synthetic set below in the table as well as the average over all five synthetic sets.

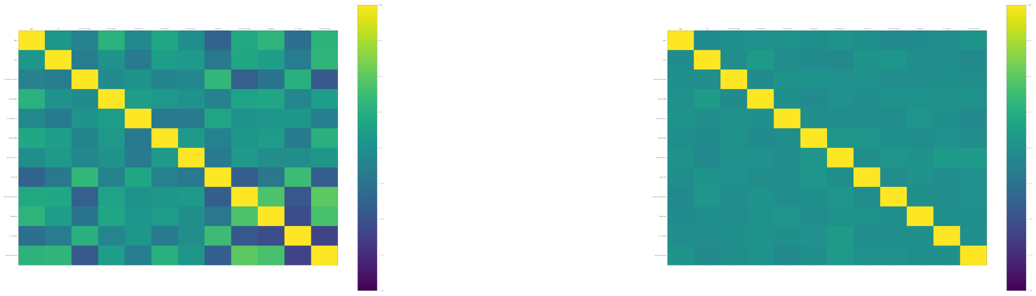
| Fold Results | | |
|---|---|---|
| Synthetic Set | Mean Test Score Over five 10-Fold Runs | Test Standard Deviation Over five 10-Fold Runs |
| Synthetic Set 1 | 0.4661 | 0.0102 |
| Synthetic Set 2 | 0.4770 | 0.0064 |
| Synthetic Set 3 | 0.4677 | 0.0158 |
| Synthetic Set 4 | 0.4889 | 0.0234 |
| Synthetic Set 5 | 0.4502 | 0.0063 |
| Average | 0.4700 | 0.0128 |

Table 6: Table 6 showing the fidelity results for each synthetic set - as well as the average - over the heart failure kaggle dataset

As these results show, there are still limitations of the current implementation of SynthVAE that need to be investigated, these are discussed in the next section.

# 7 Limitations

Upon completion of our investigations, a couple of issues have presented themselves that currently remain unfixed. Namely, we found that generating correlations between synthetic variables in SynthVAE seems to be difficult when implementing the GMM pre-processing. Good distributional metrics are shown in our results however, the correlation matrices between variables does not resemble those of the original dataset. This inevitably leads to poor downstream task performance and hence a low utility of synthetic data. An example of one such correlation matrix is shown below in Figure 6.



(a) Original heart dataset

(b) Synthetic dataset

Figure 6: Figure showing two correlation matrices for the heart dataset [17]. These correlations are found using the Spearman correlation metric

Our initial investigations into this issue surround the loss function itself. We know that TVAE and CTGAN can both capture these correlations, however it requires a careful selection of training and hyperparameters. Our loss differs to TVAE's loss as we combine the continuous and discrete log-likelihoods to form the reconstruction loss. It might be worthwhile investigating if normalisation of this loss would lead to better performance during training. The reasoning for this is as follows: if there is an imbalance between these individual losses, then the model could spend time initially focusing on matching the distributions rather than learning correlations in the data. There is also the potential problem that introducing the GMM modelling for numerical variables may alter the ability to learn correlations with variable types where GMM modelling is not applied.

# 8 Future Work and Conclusions

Due to time constraints as well as known issues, shown in the previous chapter on limitations, a deeper investigation of the fairness constraints has been left for future work. The most immediate avenue for future work should be investigations into these known errors and evaluating the normalisation/balancing of the reconstruction loss in SynthVAE. This will then allow for completion of the example use case highlighting the utility of our synthetic data generation method. A comparison to other popular methods, such as CTGAN, on the same dataset would then help identify where utilising each approach can be beneficial.

A very important area for future synthetic data generation investigations should be surrounding the fairness of data. It is desirable to be able to "de-bias" originally unfair training data. This can then help reduce the disparity within healthcare research. This would be a hugely beneficial component for SynthVAE to display. The example use case presented here has known bias within it surrounding gender and potentially age. An initial focus should be on single variable fairness mitigation as this will present itself as an easier problem. This can then be compared to other frameworks such as TabFairGAN and DECAF. An analysis of how fairness performs when using differing causal discovery methods, or the reduction in performance when having to discover the DAG (opposed to specifying it prior) could also be a useful avenue of research.

A more in-depth literature review into dealing with multiple protected variables could be useful. Mitigating for two or more variables could plausibly end up with limitations. For example, if we mitigate for age and gender in our use case we need to ensure that demographic parity is both fulfilled for age/gender separately but also for combinations of differing age and genders. To my knowledge very few papers look into this robustly and is a much needed addition to suit realistic scenarios. Finally, an extension at looking at commonly used datasets in fairness literature may be a useful avenue to allow for further comparison to other models. The COMPAS case study [18] looking at risk of re-offence from convicted criminals is a notorious dataset for fairness based ML which could be considered.

# References

[1] NHS Health Education England. 2021. AI Roadmap Methodology and findings report.
https://digital-transformation.hee.nhs.uk/binaries/content/assets/digital-transformation/dart-ed/ai-roadmap-march-2022-edit.pdf

[2] Johnson, A., Pollard, T., and Mark, R. (2016). MIMIC-III Clinical Database (version 1.4). PhysioNet.

[3] Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. Scientific Data, 3, 160035.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks

[5] Bourou, S. El Saer, A. Velivassaki, T.H. Voulkidis, A. Zahariadis, T. A Review of Tabular Data Synthesis Using GANs on an IDS Dataset. Information 2021, 12, 375.

[6] Leslie D, Mazumder A, Peppin A, Wolters M K, Hagerty A. Does "AI" stand for augmenting inequality in the era of covid-19 healthcare? BMJ 2021; 372 :n304

[7] Dominic Danks. 2021. VAEs for Synthetic Data Generation.
https://github.com/nhsx/SynthVAE/blob/main/reports/report.pdf

[8] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional GAN. CoRR, abs/1907.00503

[9] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. ACM Comput. Surv. 54, 6, Article 115 (July 2022).

[10] D. Xu, S. Yuan, L. Zhang and X. Wu, "FairGAN+: Achieving Fair Data Generation and Classification through Generative Adversarial Nets," 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 1401-1406

[11] Rajabi, Amirarsalan, and Ozlem O. Garibay. 2022. "TabFairGAN: Fair Tabular Data Generation with Generative Adversarial Networks" Machine Learning and Knowledge Extraction 4, no. 2: 488-501

[12] Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of Causal Discovery Methods Based on Graphical Models. Frontiers in Genetics. Volume 10

[13] Boris van Breugel, Trent Kyono, Jeroen Berrevoets, Mihaela van der Schaar. 2021. DECAF: Generating Fair Synthetic Data Using Causally-Aware Generative Networks. Advances in Neural Information Processing Systems 34.

[14] William A. Knaus, Frank E. Harrell, Joanne Lynn, Lee Goldman, Russell S. Phillips, Alfred F. Connors, Neal V. Dawson, William J. Fulkerson, Robert M. Califf, Norman Desbiens, Peter Layde, Robert K. Oye, Paul E. Bellamy, Rosemarie B. Hakim, and Douglas P. Wagner. 1995. The support prognostic model: Objective estimates of survival for seriously ill hospitalized adults. Annals of internal medicine, 122(3):191-203.

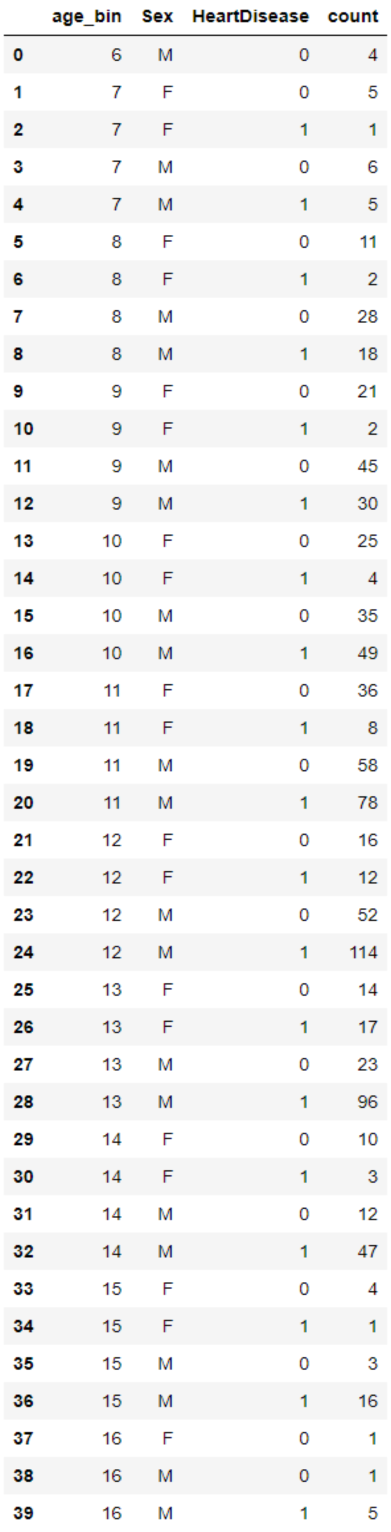
[15] Neha Patki, Roy Wedge, Kalyan Veeramachaneni. The Synthetic Data Vault. IEEE DSAA 2016.

[16] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25rd International Conference on Knowledge Discovery and Data Mining

[17] fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [May 2022] from https://www.kaggle.com/fedesoriano/heart-failure-prediction.

[18] Jeff Larson, Marjorie Roswell, Vaggelis Atlidakis. compas-analysis. 2016. Github Repository. https://github.com/propublica/compas-analysis

# Appendix A



| | age_bin | Sex | HeartDisease | count |
|---|---|---|---|---|
| **0** | 6 | M | 0 | 4 |
| **1** | 7 | F | 0 | 5 |
| **2** | 7 | F | 1 | 1 |
| **3** | 7 | M | 0 | 6 |
| **4** | 7 | M | 1 | 5 |
| **5** | 8 | F | 0 | 11 |
| **6** | 8 | F | 1 | 2 |
| **7** | 8 | M | 0 | 28 |
| **8** | 8 | M | 1 | 18 |
| **9** | 9 | F | 0 | 21 |
| **10** | 9 | F | 1 | 2 |
| **11** | 9 | M | 0 | 45 |
| **12** | 9 | M | 1 | 30 |
| **13** | 10 | F | 0 | 25 |
| **14** | 10 | F | 1 | 4 |
| **15** | 10 | M | 0 | 35 |
| **16** | 10 | M | 1 | 49 |
| **17** | 11 | F | 0 | 36 |
| **18** | 11 | F | 1 | 8 |
| **19** | 11 | M | 0 | 58 |
| **20** | 11 | M | 1 | 78 |
| **21** | 12 | F | 0 | 16 |
| **22** | 12 | F | 1 | 12 |
| **23** | 12 | M | 0 | 52 |
| **24** | 12 | M | 1 | 114 |
| **25** | 13 | F | 0 | 14 |
| **26** | 13 | F | 1 | 17 |
| **27** | 13 | M | 0 | 23 |
| **28** | 13 | M | 1 | 96 |
| **29** | 14 | F | 0 | 10 |
| **30** | 14 | F | 1 | 3 |
| **31** | 14 | M | 0 | 12 |
| **32** | 14 | M | 1 | 47 |
| **33** | 15 | F | 0 | 4 |
| **34** | 15 | F | 1 | 1 |
| **35** | 15 | M | 0 | 3 |
| **36** | 15 | M | 1 | 16 |
| **37** | 16 | F | 0 | 1 |
| **38** | 16 | M | 0 | 1 |
| **39** | 16 | M | 1 | 5 |

Figure 7: Figure showing the distribution of the kaggle heart dataset [17] with relation to its protected variables. *age_bins* here go from 6-16 and they represent age ranges in 5 year chunks from ages 25-75.