MACHINE LEARNING USING SYNTHETIC AND REAL DATA: SIMILARITY OF EVALUATION METRICS FOR DIFFERENT HEALTHCARE DATASETS AND FOR DIFFERENT ALGORITHMS

RACHEL HEYBURN [†], RAYMOND R. BOND, MICHAELA BLACK, MAURICE MULVENNA, JONATHAN WALLACE, DEBORAH RANKIN, BRIAN CLELAND

Faculty of Computing, Engineering and the Built Environment, Ulster University, Shore Road, Newtownabbey, Co. Antrim, BT37 0QB, UK

Sharing data is often a risk in terms of security and privacy especially if the data is sensitive. Algorithms can be used to generate synthetic data from an original raw dataset in order to share data that are considered more 'privacy preserving', and that increase the level of anonymity. In this paper, we carry out an experiment to study the validity of conducting machine learning on synthetic data. We compare the evaluation metrics produced from machine learning models that were trained using synthetic data with metrics yielded from machine learning models that were trained using the corresponding real data.

1. Introduction

The volume of data being generated every year is growing exponentially. A report from IBM[1] in 2013 said that 90% of the world's data was produced over the last two years and a more recent report from IBM[2] titled "10 Key Marketing Trends for 2017" said that we create over 2.5 quintillion bytes of data every day. Data scientists are availing of this huge mass of data to solve real world problems for the greater good of society and data science has already proven its worth in areas such as policing, target marketing and in new technologies like self-driving cars. We know data science also has the potential to hugely improve areas such as healthcare and cybersecurity – but why have these improvements not been observed already? The answer lies in an issue that faces many data scientists: the availability of data. Privacy concerns over health care data, for example, mean that although the data exists, it is deemed too sensitive to be available for sharing outside of specialised servers for public use. Also, in light of the forthcoming GDPR, data sharing and data use will demand careful governance. In fraud detection, instances of fraud may be so rare that there is simply not enough data to which data science techniques can be applied. Machine Learning models, for example, rely on examples of fraud from which to learn, so that when they are faced with a previously unseen set of data they can accurately predict whether something should be classed as fraudulent or not fraudulent. One way to overcome the issue of data availability is to use synthetic data rather than real data[3-5]. Synthetic data is generated from real data by using the underlying statistical properties of the real data to produce synthetic datasets which exhibit these same statistical properties. Some work has been done to ascertain whether synthetic data can preserve hidden complex patterns that data mining can uncover in the same way it would when mining the original dataset [6]. A good synthetic dataset should replace sensitive values and provide stronger guarantees of privacy and anonymity. Synthetic data can be used in two ways:

[†] Work partially supported by the European Union as part of the Meaningful Integration of Data, Analytics and Services (MIDAS) project from the European Union's Horizon 2020 programme.

.

1. To increase the size of a dataset, for times when a dataset is unbalanced due to the limited occurrence of an event.
2. To generate a full synthetic dataset that is representative of the original dataset, for times when data is not available due to its sensitive nature.

The aim of this work is to explore whether synthetic data can be a reliable replacement for real world data used by machine learning algorithms. This paper looks at ways to generate synthetic datasets and evaluates their performance when they are used to train machine learning models.

2. Methodology

2.1. Dataset Selection

For this work, synthetic datasets were generated for two datasets from the UCI Repository. The first was the Breast Cancer Wisconsin dataset which has numeric variables, 699 rows and ten attributes, plus the class attribute. Each instance belongs to one of two classes: benign, represented by a 2 in the dataset, or malignant, represented by a 4 in the dataset. The second was the Nursery dataset which has categorical variables, 12,960 rows and eight attributes, plus the class attribute. Each instance belongs to one of five classes –‘ not_recom’, ’recommend’, ’very_recom’, ’priority’ or ‘spec_prior’. It was not difficult to find data to work with for this project as the synthesis of data can be demonstrated on most datasets. However, the reason for choosing these two datasets was to determine if the variable type or the size of the dataset had any bearing on the synthesis of data. The original Breast Cancer dataset, along with the synthetic datasets subsequently generated from it, are the datasets used to train the machine learning models for which we will compare the evaluation metrics.

2.2. Generating Synthetic Data

Generating synthetic data for the purposes of balancing datasets requires the SMOTE (Synthetic Minority Over-Sampling Technique)[7] function, which uses a K-nearest neighbour algorithm, for example, to generate synthetic observations of the rare event. The SMOTE function is available in Weka and R. Python provides a module called ‘Imbalance Learn’ which has a similar function. R offers a convenient approach to generating a full synthetic dataset using a library called ‘Synthpop’ which is “a tool for producing synthetic versions of microdata containing confidential information so that they are safe to be released to users for exploratory analysis”[8]. This tool takes the variables in the dataset and, in turn, generates synthetic values using classification/regression trees or parametric models, depending on the type of variable. Synthetic data is produced using a syn() function which provides the user with control over which method should be used; either the default method or a parametric method. Two full synthetic datasets were generated for the Breast Cancer dataset using the Synthpop library. The first synthetic dataset was generated using the default method in the syn() function and the second using the ‘parametric’ method in the syn() function. The way in which synthetic data was generated for each column in the synthetic Breast Cancer datasets is shown in Table 1.1.

Table 1.1. Table showing which model was used to generate synthetic data in each column of the Breast Cancer dataset, using the ‘default’ and ‘parametric’ methods in the syn() function.

	Default	Parametric
Sample Code #	Sample	Sample
Clump Thickness	Cart	Norm Rank
Uniformity of Cell Size	Cart	Norm Rank
Uniformity of Cell Shape	Cart	Norm Rank
Marginal Adhesion	Cart	Norm Rank
Single Epithelial Cell Size	Cart	Norm Rank
Bare Nuclei	Cart	Polyreg

Bland Chromatin	Cart	Norm Rank
Normal Nucleoli	Cart	Norm Rank
Mitoses	Cart	Norm Rank
Class	Cart	Norm Rank

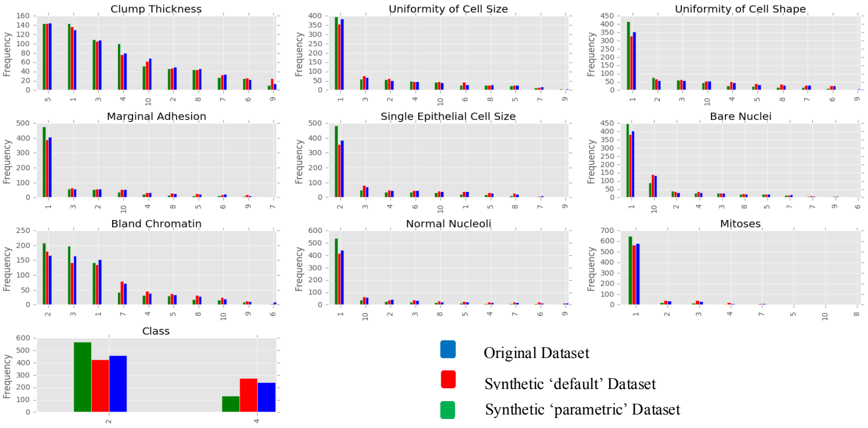


Figure 1.1. Figure showing how the distributions of each variable in the two synthetic Breast Cancer datasets compare to those in the original Breast Cancer dataset

In both methods, the synthetic unique identifiers 'sample code number' are generated using a random sample from the observed data. In the default method the rest of the synthetic variables are generated by drawing from conditional distributions fitted to the original data using classification/regression tree models.[6] In the 'parametric' method, the synthetic values are found using 'normrank': normal linear regression preserving the marginal distribution and 'polyreg': unordered polytomous regression. Figure 1.1 shows how the distributions of the two synthetic Breast Cancer datasets look compared to the original dataset. Observing the distributions of both synthetic Breast Cancer datasets, it is clear that they exhibit similar underlying statistical properties as that of the original dataset. In addition, this project involved the development of a new method to generate synthetic data for the Breast Cancer dataset. It used the underlying distributions of each variable and a machine learning algorithm to generate the new synthetic values. This involved randomly sampling the unique identifier 'sample code number'. Then, for each column in the dataset, apart from the last, we determined the weight of each value within the column and used the random.randint() function in Python to generate a new synthetic column of values that was representative of the original column. This provided a synthetic dataset containing all but the class variable. To determine this class variable, a decision tree classifier was trained using the original dataset and used it to predict the class of each instance in my synthetic dataset. The distribution of this synthetic dataset compared to the original Breast Cancer dataset is shown in Figure 1.2.

To understand whether the type of variable impacts the synthesis of data in the Synthpop library in R, the two methods in the syn() function was used to generate synthetic data for the categorical Nursery dataset. Table 1.2 shows how the synthetic data was generated for each column in both of the synthetic datasets.

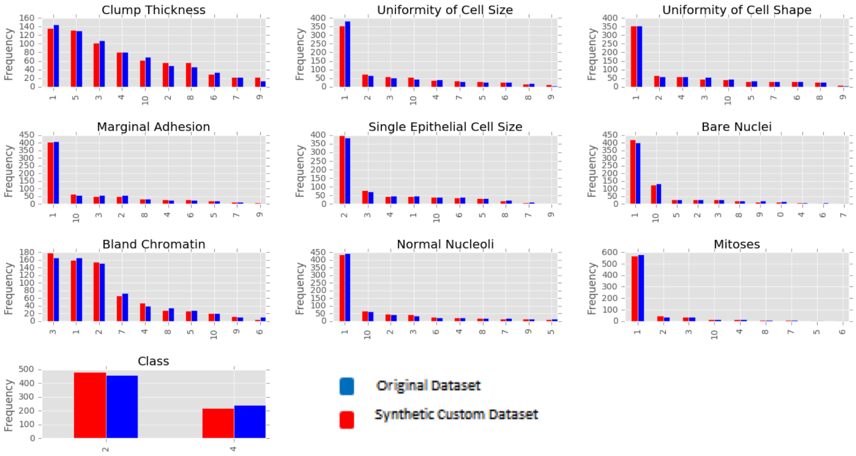


Figure 1.2. Figure showing how the distributions of each variable in the third synthetic Breast Cancer dataset compared to those in the original Breast Cancer dataset

Table 1.2 Table showing which model was used to generate synthetic data in each column of the Nursery dataset, using the 'default' and 'parametric' methods in the syn() function.

	Default	**Parametric**
Parents	Sample	Sample
Has_Nurs	Cart	Polyreg
Form	Cart	Polyreg
Children	Cart	Polyreg
Housing	Cart	Polyreg
Finance	Cart	Logreg
Social	Cart	Polyreg
Health	Cart	Polyreg
Class	Cart	Polyreg

Synthetic data is generated identically for both the Breast Cancer dataset and the Nursery dataset using the 'default' method in the syn() function. Using this method, we see that categorical or numerical variables have no bearing on how the synthetic data is generated. However, we see a difference when we use the 'parametric' method. In the numerical Breast Cancer dataset, most of the synthetic variables are generated using the normal linear regression, with one generated using polytomous (multinomial) regression. In the categorical Nursery dataset, most synthetic variables are generated using polytomous regression and one generated using logistic regression. Figure 1.3 shows how the distributions of the two synthetic Nursery datasets look compared to the original dataset.

We observe that the distributions of both synthetic Nursery datasets are almost identical to the distribution of the original dataset and can observe that the statistical properties of the original dataset have been preserved. We also observe that the type of variable or the size of the dataset has no bearing on the distributions of synthetic data which has been generated using the syn() function in R.

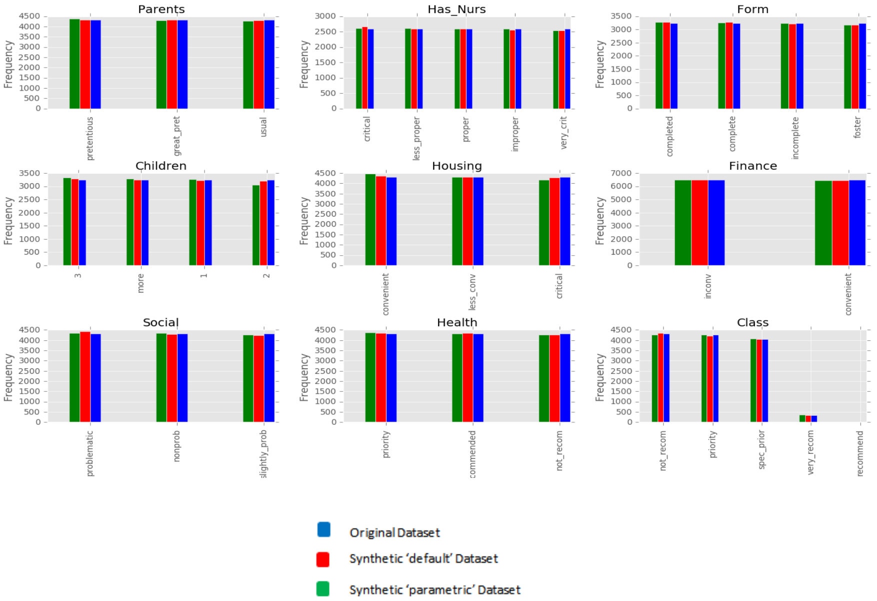


Figure 1.3 Figure showing how the distributions of each variable in the two synthetic Nursery datasets compare to those in the original Nursery dataset

2.3. Machine Learning Using Real and Synthetic Data

To evaluate whether synthetic datasets can be used in place of real datasets in machine learning models, different classification models were trained with the original and synthetic Breast Cancer datasets. For this part of the work Python's machine learning library, Scikit – Learn was used, as it has a wide selection of algorithms and a consistent API. For the Breast Cancer dataset with binary classification a Linear Classification model, a Decision Tree Classifier, a K-Nearest Neighbour Classifier, a Support Vector Machine Classifier and a Random Forest Classifier were used. A combination of simple and more complex algorithms was purposively chosen to see how well each model performed when trained with the synthetic and real datasets. For training and testing, 10-fold cross validation (CV) was used, as this is a more sophisticated holdout training and testing procedure than simply splitting the data into one training and test set, and makes better use of limited data. To implement the Linear Classification, the SGDClassifier with default parameters and loss=*'hinge'* and *random_state= 0 were used.* To implement the Decision Tree Classification the DecisionTreeClassifier with default parameters and *criterion='gini'*, *max_depth=10*, *random_state=0 were used.* To implement the k-Nearest Neighbour Classifier the KNeighborsClassifier with default parameters with leaf_size=30, metric='minkowski', n_jobs=2, n_neighbors=10, p=2 and weights='uniform' were used. To implement the Random Forest Classifier, the RandomForestClassifier with default parameters and *n_estimators=10*, *criterion='gini'*, *max_depth=10*, *min_samples_split=2* and *random_state= 1 was used.* Finally, to implement the Support Vector Machine Classifier, SVC with default parameters and *C=1.0*, *kernel='rbf'*, *degree=3*, *probability=True* and *random_state=None were used.* The parameters in the Machine Learning models were the same when training the original Breast Cancer dataset and the three synthetic Breast Cancer datasets, to enable precise comparison of the evaluation metrics.

3. Results

To compare the performances of each model after being trained with the original and synthetic datasets, a variety of evaluation metrics were used. The first, most obvious evaluation metric to compare was the accuracy of each model. Table 1.3

compares the accuracy each model achieved after being trained by the three datasets.

Table 1.3. Table comparing the accuracy scores achieved by each model as trained by each dataset

Dataset	**Linear Model**	**Decision Tree**	**KNN**	**Random Forest**	**SVM**
Original	0.971428	**1.0**	0.969999	0.998571	0.995714
Default Synthetic	0.922753	**0.997142**	0.942795	0.989999	0.99
Parametric Synthetic	0.894161	**0.998571**	0.952836	0.989999	0.985714
Custom Synthetic	0.6882194	0.998550	0.864099	**0.998571**	**0.998571**

We see that the most accurate model for both the original and default and parametric synthetic datasets is the Decision Tree. It achieves a perfect accuracy score when trained with the original dataset and very high accuracy for each of the synthetic datasets. However, the synthetic dataset which was trained using the 'parametric' method performs slightly better. The most accurate models for the custom synthetic dataset are the Random Forest and SVM, followed closely by the Decision Tree. The least accurate model for the original dataset was the k-Nearest Neighbour classifier, while for the three synthetic datasets the least accurate model was the Linear Model. The Linear Model also provides the largest variation in accuracy score between the four datasets. We observe that the accuracy score in all other models does not vary significantly across the four datasets. Accuracy can often be too simplistic, so it is vital that we use other evaluation metrics to fully understand how the models are performing. Precision scores, recall scores and the F1 measure evaluation metrics should provide more insight into model performance. These evaluation metrics are shown in Table 1.4, Table 1.5 and Table 1.6.

Table 1.4. Table comparing the precision scores of each model after being trained by each dataset

Dataset	**Linear Model**	**Decision Tree**	**KNN**	**Random Forest**	**SVM**
Original	0.972	1.000	0.970	0.999	0.996
Default Synthetic	0.930	0.997	0.943	0.990	0.990
Parametric Synthetic	0.899	0.999	0.954	0.990	0.986
Custom Synthetic	0.786	0.999	0.873	0.999	0.984

Table 1.5. Table comparing the recall scores of each model after being trained by each dataset

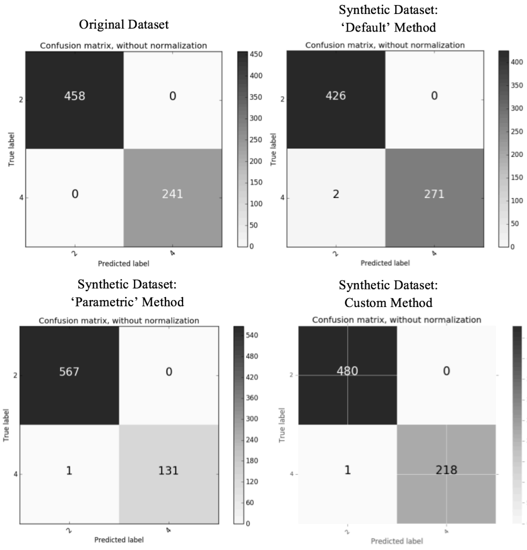
Dataset	**Linear Model**	**Decision Tree**	**KNN**	**Random Forest**	**SVM**
Original	0.971	1.000	0.970	0.999	0.996
Default Synthetic	0.923	0.997	0.943	0.990	0.990
Parametric Synthetic	0.894	0.999	0.953	0.990	0.986
Custom Synthetic	0.688	0.999	0.864	0.999	0.984

Table 1.6. Table comparing the F1 scores of each model after being trained by each dataset

Dataset	**Linear Model**	**Decision Tree**	**KNN**	**Random Forest**	**SVM**
Original	0.971	1.000	0.970	0.999	0.996

Default Synthetic	0.923	0.997	0.943	0.990	0.990
Parametric Synthetic	0.879	0.999	0.951	0.990	0.985
Custom Synthetic	0.562	0.999	0.855	0.999	0.984

We see that precision, recall and F1 scores for each model for each dataset offer the same insight into model performance as the accuracy score. In terms of these evaluation metrics, the Decision Tree is still the best classifier for all datasets, with the Random Forest also performing well for the custom synthetic dataset. We observe that the Linear Model provides the largest variation in precision, recall and F1 scores between the four datasets. Although precision, accuracy and F1 measures are summaries of the confusion matrix in some form; it is still beneficial to separate out the decisions made by the model, to show where one class is being misclassified for another. Figure 1.4 shows the confusion matrices for the Decision Tree, trained by the original dataset and the three synthetic datasets.



Original Dataset

Synthetic Dataset: 'Default' Method

Synthetic Dataset: 'Parametric' Method

Synthetic Dataset: Custom Method

Figure 1.4 Confusion matrices for the Decision Tree Classifier after being trained by each dataset

Figure 1.4 shows that the Decision Tree Classifier correctly predicted every instance in the original dataset. The Decision Tree Classifier for each synthetic dataset both incorrectly predicted the class as being a '2' when in fact its true class was a '4'; however the 'default' synthetic dataset did this twice as often as the 'parametric' and custom datasets. This means that twice in the 'default' synthetic dataset a tumour was predicted to be non-cancerous when in fact it was cancerous. This highlights the importance of using more evaluation metrics than just an accuracy score. The accuracy score makes no distinction between false positive and false negative errors and makes an assumption that they are equally important. This kind of misclassification could be very dangerous in the real world; classifying a tumour as non-cancerous when the tumour is cancerous is more serious than classifying the tumour as cancerous when it is non-cancerous. Therefore, while the synthetic datasets can be a very close match to the original data in terms of their distributions and is a feasible solution to producing a dataset that enhances privacy, when the class variable has such high importance, we need to err on the side of caution if we wish to use them to train machine learning models. We also need to ensure the use of multiple evaluation metrics, not just the simple accuracy score which can be misleading in terms of how the model is actually performing. In this instance, the use of synthetic data to predict whether a tumor is cancerous or non-cancerous may not be recommended as the consequences to classifying an instance incorrectly are too serious.

4. Conclusion

In this very early work, we can see that the evaluation metrics achieved when machine learning (training and testing) using synthetic data are similar to the evaluation metrics achieved when machine learning (training and testing) using the real datasets. That is in terms of accuracy, precision, recall and F1 scores. In this limited case study in this paper, the model performance when trained and tested using synthetic data was similar to the performance of the model that was trained and tested with the real data. This is only one case study, but this may suggest that the evaluation of models built using synthetic data maybe reflective of the results that would be achieved if real data had of been used. If further research supports this hypothesis, then data scientists can mine synthetic healthcare datasets with an assumption that any knowledge elicited is very likely to be reflected in the real dataset. This could open up competitions and health data mining to more data scientists. Using synthetic datasets to facilitate privacy preserving machine learning to discover patterns and viable predictive modelling without giving away raw sensitive data maybe a useful process to minimize risk. We recognize that this work is primitive and limited since there was no cross-testing of the models. Future work would include testing a machine learning model that was built using synthetic data on real data to ascertain if a model trained using synthetic data would perform just as well with real world cases.

Acknowledgments: We acknowledge support from the 'Meaningful Integration of Data, Analytics and Services (MIDAS)' project as funded by the European Union's Horizon2020 programme-H2020-SC1-2016-CNECT SC1-PM-18-2016-Big Data Supporting Public Health Policies, Grant Agreement No. 727721

5. References

[1]: www-01.ibm.com. (2018). *IBM What is big data? Bringing big data to the enterprise - India*. [online] Available at: https://www-01.ibm.com/software/in/data/bigdata/ [Accessed 29 Jan. 2018].

[2] www-01.ibm.com. (2018). *10 Key Marketing Trends for 2017*. [online] Available at: https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN [Accessed 29 Jan. 2018].

[3] C.C. Aggarwal, and S.Y. Philip, A general survey of privacy-preserving data mining models and algorithms. In *Privacy-preserving data mining*, Springer, Boston, MA, pp. 11-52 (2008).

[4] X. Qi, and M. Zong, An overview of privacy preserving data mining. *Procedia Environmental Sciences*, *12*, pp.1341-1347 (2012)

[5] Y.A.A.S Aldeen, M. Salleh, and M.A. Razzaque, A comprehensive review on privacy preserving data mining. *SpringerPlus*, *4*(1), p.694 (2015)

[6] J. Eno, CW. Thompson, Generating synthetic data to match data mining patterns. IEEE Internet Computing, 12(3), pp. 78-82 (2008)

[7] NV. Chawla, KW. Bowyer, LO. Hall, WP. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-57 (2002)

[8] Cran.r-project.org. (2018). *CRAN - Package synthpop*. [online] Available at: https://cran.r-project.org/web/packages/synthpop/index.html [Accessed 29 Jan. 2018].