The International Young Researchers' Conference
DOI: 10.34614/iyrc2021f27

Escalating the Quantity of Medical Data Using CTGAN: Diabetes Dataset

Jihyung Kim
jihyungkim09@gmail.com

Abstract

The number of diabetes diagnoses is increasing sharply in the United States. It is a life-long disease that can cause serious symptoms such as blurred visions. Collecting medical data requires a consent form and goes through complicated procedures, which makes it harder. Conditional Generative Adversarial Network(CTGAN) can help to solve this problem. GAN is a Deep Learning model that manufactures synthetic data. CTGAN is basically GAN because it goes through very similar procedures, but CTGAN is for table data. We checked how accurate the fake data was to the real data using various machine learning models and deep learning. Logistic Regression(LR), Decision Tree(DT), KNN, Gradient Boosting(GB), Light Gradient Boosting Machine(LGBM), Support Vector Classifier(SVC), Gaussian, and Deep Neural Network(DNN) got 40.55%, 38.1%, 44.5%, 39%, 35.35%, 44.05%, 53.65%, 39.25%, and 34.2%, respectively. We applied GridSearch on two models: Random Forest(RF) and Light Gradient Boosting Machine(LGBM). Random Forest(RF) showed a bit better accuracy by performing 77.85% while Light Gradient Boosting Machine(LGBM) performed 76.65%. Then we decided to create a new dataset combining the fake data with a bit of real data. When we compared the new dataset with the pure real data, the accuracy scores from all models almost doubled. Although we had to modify the model in order to reach a satisfactory result, CTGAN can become a very significant model for researchers who need a large amount of data.

Introduction

Background

Diabetes is a disease caused by a lack of insulin. There are two types of diabetes. Type-1 diabetes is when the body can't produce a sufficient amount of insulin by nature. Type-2 diabetes is when the body refuses to produce insulin. Type-2 is usually caused because of being inactive or overweight(Type 2 Diabetes, 2007). Most people are not careful of diabetes because they don't notice the critical symptoms of diabetes. These are the main symptoms: extreme thirst and hunger, sudden weight loss, and tiring fatigue. And if the diabetes patient gets Diabetic Retinopathy(DR), the patient could also have vision loss(Barhum, 2019). These days, many medical companies use Artificial Intelligence software to diagnose diseases such as diabetes. Figure 1 shows how the revenue from AI is increasing exponentially, which means the usage of AI technology is increasing in the world(Statista, 2020).

Objective

Medical data are very hard to collect because they need consent forms about their human rights in order to collect their data. In numerous medical research, the researchers have a hard time looking for better and more data because most of the research requires a dataset with an excessive amount of rows(Why It's so Hard for Patients to Access Their Medical Records, 2019). In order to solve the problem, we used a conditional tabular generative adversarial network (CTGAN) to duplicate and increase the data in the diabetes

dataset. Cheon et al. used CTGAN on electroencephalography(EEG) data, and could only achieve 49.8% as the highest accuracy score(Cheon et al., 2021). So we decided to use CTGAN on a diabetes dataset to see if it performs better. CTGAN first learns about the data and then makes fake data. To see if the fake data is accurate, we used seven different machine models and a default deep learning model. Our research goes in the following order: table evaluator, accuracy between the pure fake data and the original data, and accuracy between the fake data combined with original data and the original data.

Prior Research

For our prior research, we found out important factors of getting diabetes using Decision Tree (DT), Random Forest (RF), K- Nearest Neighbors (KNN), Logistic Regression (LR), Gradient Boosting (GB), XG Boosting (XGB), and Catboost(CB). We used the same data as this research, and it was hard to trust the accuracy score since it had only 2000 rows. So we decided to find a way to increase the number of rows.

Related works

Kevin Kuo utilized the CTGAN model to generate synthetical tabular data about the insurance dataset. Open-source R interface was used to evaluate the performance of the generated dataset, such as machine learning efficacy, distribution of variables, stability of model parameters, and it showed high ML efficacy on the insurance dataset. As the insurance data is not publicly available because of privacy issues, these results show the synthetical data of the insurance datasets made from CTGAN could be used in the future(Kuo, 2019).

Chen et al. attempted to generate text via customizable conditional text generative adversarial network. After constructing the model, they adopted an automated word-level replacement strategy in order to extract the specific keywords from the synthetic text. Lastly, a comprehensive evaluation metric, also known as a mixed evaluation metric was applied to compare the generated one to the real one. The proposed model achieved higher performance compared to the other existing text generation models(Chen et al., 2020).

Moon et al. focus on load forecasting which is a critical issue of a smart grid. Therefore, for better prediction, machine learning and deep learning methods have been applied, but if there exists an insufficient dataset, acquiring higher performance is difficult. The research consists of two different stages. The first stage is generating the synthetic data through the various generation models including vanilla GAN, CGAN, WGANGP, CTGAN, MTD, and TVAE. In the second stage, they created the dataset based on deep learning regression models. Lastly, they analyzed the performance of the synthetic dataset through MLP, and MAPE, RMSE, MAE were used for the evaluation(Moon et al., 2020).

Seunghyun Park and Hyun-hee Park applied an oversampling and undersampling method to the network traffic data. As the network traffic mainly consists of the normal data and a minor amount of attack data, an oversampling method is essential for classifying the network traffic. They suggested a combined oversampling and undersampling method based on the slow-start(COUSS) algorithm and it outperformed the other methods including SMOTE, borderline SMOTE, adaptive synthetic sampling, and GAN by improving the F! Scores by 8.639%, 6.858%, 5.003%, and 4.074%, respectively(Park & Park, 2020).

Cheon et al. applied CTGAN and GAN algorithms to generate the synthetic data of the EEG data. As the EEG data is difficult to gather, data augmentation is required for the BCI research. The EEG dataset was CSV format, therefore they utilized the GAN models which are suitable for the tabular data. The experiment consisted of 3 stages, in the first stage, they generated the synthetic data from each model and compared

them via visualization. Then, a table evaluator function was applied to calculate the similarity score. Lastly, each generated dataset was used as input data of the various machine learning algorithms for the classification. Even though the visualization and similarity score showed CTGAN outperform TGAN, the final stage proved that there exists no significant difference between them (Cheon et al., 2021).

Materials and Methods

Data description

We used a diabetes dataset collected from Kaggle. The data are collected from the hospital in Frankfurt, Germany. Our data has 2,000 rows, which represents the number of people, and 9 columns with "Outcomes" inclusive, which represents the number of features. The features were: pregnancy term in weeks, amount of glucose in their body, their blood pressure, skin thickness, amount of insulin in their body, body mass index(BMI), diabetes pedigree function, and age. The range of pregnancy term was 17(from 0 to 17 weeks), glucose was 199(0 to 199), blood pressure was 122(0 to 122), skin thickness was 110(0-110), insulin was 744(0-744), BMI was 80.6(0 to 80.6), and Diabetes Pedigree Function was 2.34(0.08 to 2.42). The age group of the participants of the data was adults from 21 to 81(Diabetes, 2018).

Generative Adversarial Network (GAN)

A generative adversarial network(GAN) is a deep learning model which generates synthetic data. GAN consists of a generator and a discriminator. The generator gets a random vector as an input and then generates the image. Discriminator discriminates against the image whether it is fake or real. The main purpose of the Generator is to maximize the probability of the Discriminator misjudging the image. On the contrary, the discriminator tries to minimize the probability of making a mistake, which leads to minimax problems between them. However, GAN includes some downsides, which are model collapse, non-convergence, diminished gradient, and lack of a proper evaluation metric(Creswell et al., 2018).

CTGAN

Lei Xu and Kalyan Veeramachaneni introduced a tabular generative adversarial network(TGAN) for applying the GAN to tabular data(XU & Veeramachaneni, 2018). However, as the tabular data consists of numerical variables and categorical variables, preprocessing both data types requires more time and memory. Therefore, a conditional tabular generative adversarial network(CTGAN) was introduced by the same researchers, to suggest better-preprocessing methods, especially for highly imbalanced categorical columns. TGAN uses a gaussian mixture model (GMM) for the training, which aims to make a distribution with a weighted sum of m Gaussian distributions. However, CTGAN utilizes a variational gaussian mixture model (VGM) instead of GMM to deal with numerical variables. CTGAN models the distribution of columns with VGM. Then, for each value, CTGAN computes the probability of each model. Lastly, it samples a mode and normalizes the value. As the TGAN implies a limitation of "class imbalance" in categorical variables, a conditional vector, loss, and "training-by-sampling" are used to solve the downside. The discrete columns usually reshape into one-hot vectors, therefore, for a more efficient preprocessing, CTGAN transfers them into mask vectors. Generator loss imposes a penalty on its loss by adding cross-entropy(Xu et al., 2019).

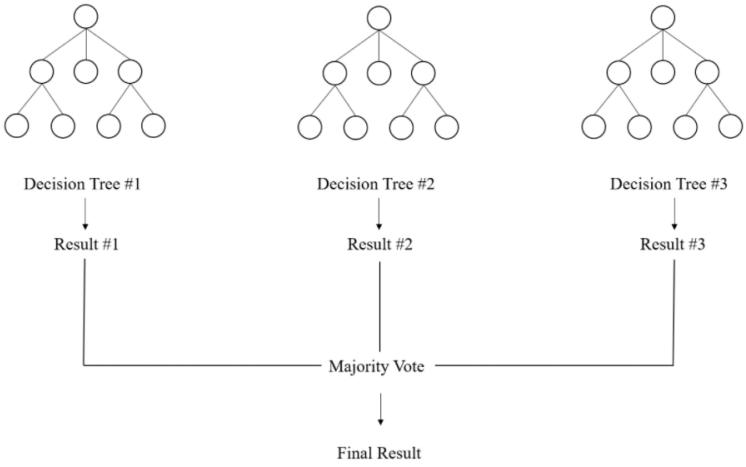
Random Forest

Random forest is a representative ensemble algorithm in machine learning(Breiman, 2001). Ensemble algorithm refers to a technique that produces multiple classifiers and combines them to produce more accurate predictions. Instead of using a single powerful model, a combination of several weaker models helps predict or classify more accurately(Dietterich, 2000). The decision tree is used as a classifier in the random forest. The bagging method is about extracting the mini

dataset from the original data and then using them as input data for each classifier and also yields parallel computation. This extracting method is called bootstrapping and it allows redundancy in sampling. After each decision tree classifies the mini dataset, a final result is decided through the majority votes (Garboden, 2019).

Results

At first, we tried to find the accuracy scores between the fake data made by CTGAN and the default data. But as Figure 10 shows, the accuracy scores as the results were too low:



Logistic Regression(LR), Decision Tree(DT), KNN, Gradient Boosting(GB), Light Gradient Boosting Machine(LGBM), Support Vector Classifier(SVC), Gaussian, and Deep Neural Network(DNN) got 40.55%, 38.1%, 44.5%, 39%, 35.35%, 44.05%, 53.65%, 39.25%, and 34.2%, respectively. So we decided to try two different ways to improve the accuracy scores. One of them was to use GridSearch to find the optimal(hyper) parameter. We used GridSearch on Random Forest(RF) and Light Gradient Boosting Machine(LGBM), and got 77.85% and 76.65%, respectively. Another method to increase the accuracy was by combining the fake data with a bit of real data. When we compared the fake data combined with a bit of real data to the real data, Random Forest(FR) achieved the best accuracy score, getting a 100%. Other machine learning and deep learning machines also performed decent results. As Figure 11 shows, Logistic Regression(LR), Decision Tree(DT), KNN, Gradient Boosting(GB), Light Gradient Boosting Machine(LGBM), Support Vector Classifier(SVC), Gaussian, and Deep Neural Network(DNN) got 77.3%, 88.5%, 93.9%, 93.05%, 88%, 93.9%, 75.05%, and 65.8%, respectively. Figure 5, Figure 6, Figure 7, Figure 8, Figure 9 shows the comparison between the fake data combined with a bit of real data to the real data. Figure 5 shows how the mean and standard deviation between the data are almost identical exhibiting a linear line with a slope of 1. Figure 6 shows the cumulative sum of each feature. The orange lines and the blue lines each represent fake and real data, and in each graph, it is hard to see the blue(real) line because they are overlapping for most of the parts. That means they are very similar. A similar thing is happening with Figure 7, too. For the most part, each bar graphs overlap. Although there are some erratic peaks(in the graphs of pregnancies and blood pressure), the basic frames of the graphs are almost uniformed. Figure 8 shows the correlation map(heat map) of real data and fake data, and another heat map of the differences between each correlation map. The heat map that shows the differences has only a few light red colors on it, which means they are very similar. The scatter plots shown in Figure 9 are almost indistinguishable. Even the outliers are in similar locations. These figures and graphs were very helpful to understand how similar the data were.

Discussions

Principal Finding

The accuracy scores CTGAN performed were very high when we combined our fake data with a bit of the original data. Although Gan is not a popular model, researchers can use the model to increase the number of data when they lack it. Many researchers in the medical industry have a hard time collecting valid information and data because consent forms are required in order to do so. However, modification is needed for Gan since the fake data themselves weren't as accurate.

Limitation

As mentioned, the pure fake data CTGAN created wasn't as accurate. The accuracy score was only high when we combined the data with a bit of real data. Therefore, we can't say that the model itself is perfect. For our further research, we will research more about GANs and modify this problem.

Conclusion

The purpose of our research is to create a synthetic diabetes dataset, in order to solve the problem of collecting medical data. We applied CTGAN to the given dataset as the format of our dataset is a CSV file. Then, we evaluated our synthetic data and real data through a table evaluator function. In addition, various machine learning and deep learning methods were used to get the accuracy score. However, the accuracy score was lower than we expected, therefore, we combined the synthetic data and real data and used them as input data. The result showed that Random Forest with Grid Search showed an accuracy score of 100 %. To sum up, even though synthetic data alone could not yield high performance, we can achieve better performance through combining the real one and the synthetic one. It showed the potential to resolve the difficulty of obtaining medical data. Therefore, for further research, we will focus on achieving higher performance by the synthetic data alone.

References

Breiman, L. (2001). Random Forests. Machine Learning. Published. https://doi.org/10.1023/A:1010933404324

Briefing, D. (2019). Why it's so hard for patients to access their medical records. Advisory Board. Published. https://www.advisory.com/daily-briefing/2019/09/10/medical-records

Chen, J., Wu, Y., Jia, C., Zheng, H., & Huang, G. (2020). Customizable text generation via conditional text generative adversarial network. Neurocomputing. Published. https://doi.org/10.1016/j.neucom.2018.12.092

Cheong, M. J., Lee, D. H., Park, J. W., Choi, H. J., Lee, J. S., & Lee, O. (2021). CTGAN VS TGAN? Which one is more suitable for generating synthetic EEG data. Journal of Theoretical and Applied Information Technology, 99(10). http://www.jatit.org/volumes/Vol99No10/15Vol99No10.pdf

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative Adversarial Networks: An Overview. IEEE Signal Processing Magazine, 35(1). https://doi.org/10.1109/msp.2017.2765202

Dansinger, M. (2020). Type 2 Diabetes. Type 2 Diabetes. Published. https://www.webmd.com/diabetes/type-2-diabetes

Dataset of Diabetes. (2018). Kaggle. https://www.kaggle.com/johndasilva/diabetes

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. Multiple Classifier Systems, 1857. https://doi.org/10.1007/3-540-45014-9_1

Garboden, P. M. E. (2020). Sources and Types of Big Data for Macroeconomic Forecasting. Macroeconomic Forecasting in the Era of Big Data, 52, 3–23. https://doi.org/10.1007/978-3-030-31150-6_1

Kuo, K. (2019). Generative Synthesis of Insurance Datasets. ArXiv Preprint ArXiv:1912.02423. Published. https://arxiv.org/abs/1912.02423

Moon, J., Jung, S., Park, S., & Hwang, E. (2020). Conditional Tabular GAN-Based Two-Stage Data Generation Scheme for Short-Term Load Forecasting. IEEE Access, 8. https://doi.org/10.1109/ACCESS.2020.3037063.

Park, S., & Park, H. (2021). Combined oversampling and undersampling method based on slow-start algorithm for imbalanced network traffic. Computing 103. Published. https://doi.org/10.1007/s00607-020-00854-1

Revenue from artificial intelligence systems in healthcare worldwide from 2013 to 2021. (2018). Statista. https://www.statista.com/statistics/938775/global-healthcare-artificial-intelligence-market-revenue/

Weatherspoon, D. (2019). What are the symptoms of type 2 diabetes? Medical News Today. Published. https://www.medicalnewstoday.com/articles/317462

Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling Tabular data using Conditional GAN. ArXiv Preprint ArXiv:1907.00503. Published. https://arxiv.org/abs/1907.00503

Xu, L., & Veeramachaneni, K. (2018). Synthesizing Tabular Data using Generative Adversarial Networks. ArXiv Preprint ArXiv:1811.11264. Published.