MACHINE LEARNING WITH DATA AUGMENTATION TO PREDICT GLUCANTIME EFFECTIVENESS AGAINST CUTANEOUS LEISHMANIASIS

Juan José Hoyos Urcué
Javeriana Cali University
Engineering and Sciences Faculty
Cali, Colombia
juanjohupuj@javerianacali.edu.co

ABSTRACT

Facing data analysis problems on small data sets is a common problem in medical research; likewise, it is a problem that makes the application and success of classic machine learning algorithms very difficult. Many techniques have tackled the problem of a small data set, mainly for computer vision and image processing fields. However, for tabular data, short has been disseminated. In this degree project, the use of tabular data augmentation techniques is proposed to introduce synthetic instances quite similar to real instances, particularly in the context of a medical/social problem of predicting the effectiveness of Glucantime as a treatment against cutaneous Leishmaniasis. Experiments show that using these data augmentation algorithms enhances the characteristics of the initial data set and improves the performance of machine learning models. The dataset used in this investigation has ten attributes and 18 registers.

Keywords: Machine Learning, Tabular Data Augmentation, Cutaneous Leishmaniasis, Infectious Disease, Synthetic Data, Small Dataset, K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Neural Networks, Random Forest, SMOTE, Borderline SMOTE, Gaussian Mixture Model, ADASYN, Genetic Algorithm, Generative Adversarial Network.

1. INTRODUCTION

Leishmaniasis is an infectious disease transmitted by the bite of the Phlebotomine sandfly or female simulid mosquito. In Colombia, 95% of the annual cases are cutaneous Leishmaniasis [1]. The most widely used treatment to treat this disease is Glucantime, a highly toxic chemical substance that can generate severe adverse effects and even worse than the disease itself [2]. The most worrying thing is that it does not work in 100% of patients; that is why it arose the need to study the medical history and try to predict early whether this treatment will work in patients with this disease.

The research was developed with the International Center for Medical Training and Research (CIDEIM), which provided a demographic dataset for data analysis and machine learning experiments. However, this data set was too small (ten attributes and 18 registers), which made the conventional process of experiments with machine learning difficult. For this reason, it was purposed to use data augmentation techniques to take full advantage of the real data characteristics, generating more data to train the machine learning models and get better models' performance.

All this allowed experimentally to build applicable models by health professionals, helping to decide when to use Glucantime as a treatment for their patients suffering from Leishmaniasis, positively impacting their life quality and reducing the indifference and underestimation suffered by communities affected from it.

2. LITERATURE REVIEW

2.1 Data Augmentation

ADASYN: "The essential idea of ADASYN is to use a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn compared to those minority examples that are easier to learn." [3]

SMOTE: "To create the new synthetic minority class instances, SMOTE first selects a minority class instance ***a*** at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors ***b*** at random and connecting ***a*** and **b** to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances ***a*** and **b**." [4]

Borderline SMOTE: This algorithm only oversample the minority class examples that are on the decision border. First, the examples of the decision boundary are discovered; then, synthetic examples are generated from them and added to the original set. [5]

Gaussian Mixture Model (GMM): The Expectation-Maximization algorithm is used to fit the GMM to the data set. GMM learns the representation of a multimodal data distribution as a combination of unimodal distributions. GMM fits the K Gaussian components to the data set by parameterizing each group's weight, mean, and covariance. After fitting the data with multiple Gaussian distributions, the results can be used to group any new data points into one of the identified clusters. [6]

Genetic Algorithm (GA): "Implementing GA for synthetic data generation involves random searching in a solution space of possible datasets based on optimization criteria (which are properties of the original dataset) expressed as a fitness function. (. . .) This means that we can explore different fitness function permutations to establish which are sufficient for a full set of analytical properties to emerge." [7]

Generative Adversarial Networks (GANs): "GANs often comprise a generator and a discriminator that learn simultaneously. The generator tries to capture the potential distribution of real samples and generates new data samples. The discriminator is often a binary classifier, discriminating real samples from the generated samples as accurately as possible. Both the generator and the discriminator can adopt the structure of currently popular deep neural networks. The optimization process of GANs is a minimax game process, and the optimization goal is to reach Nash equilibrium, where the generator is considered to have captured the distribution of real samples." [8]

2.2 Machine Learning

K-nearest neighbors (KNN): This algorithm has its foundation in finding the k nearest neighbors (calculating a measure of distance) of the sample to be classified, in such a way that each one of them contributes a vote for the class it belongs to, classifying the sample in the class that has the most votes. [9]

Random Forest: "Random Forest is a group of un-pruned classification or regression trees made from the random selection of samples of the training data. Random features are selected in the induction process. Prediction is made by aggregating (majority vote for classification or averaging for regression) the predictions of the ensemble." [10]

Logistic Regression: Like linear regression, the logistic regression model computes the weighted sum of the input characteristics, plus a bias term. However, instead of returning the direct result of that operation, it applies a logistic transformation that is essentially an S-shaped curve that returns a number between 0 and 1, which is defined as:

$$f(t) = \frac{1}{1 + e^{-t}} \qquad (1)$$

[11]

Support Vector Machines (SVM): This classification model builds a multidimensional optimal hyperplane to separate two classes and

minimize error. It can be easily extended to complex instances that are not linearly separable by mapping the training samples to a larger dimension space where they possibly become linearly separable by using a kernel function [12, p. 256]. Minimizing the error means making the margin as grater as possible so that the decision bounder has the maximum distance.

Multilayer perceptron: The network can contain many intermediate layers between the input and output layers, called hidden layers, which are extremely useful for modeling more complex relationships between input and output variables.

These layers perform computations that are transmitted from layer to layer until reaching the output layer. The use of an activation function is required, as in the simple perceptron. However, the most usual for a multilayer perceptron is to use activation functions other than the sign or linear function since this prevents layers from computing linear combinations of the input parameters. The goal of the neural network learning algorithm is to find a set W that represents the weight of each of the edges of the network in such a way that the total sum of the squared errors is minimized. The backpropagation algorithm solves this optimization problem efficiently. [12, p.246]

2.2.1 Evaluation Metrics

Precision: This metric allows knowing the proportion between the actual positive samples and those that the classifier predicted as positive.

Sensitivity or Recall: This metric allows knowing the proportion of samples that the classifier predicted as positive over those whose actual label was positive.

F1 Score: This metric is a harmonic mean of Precision and Recall that allows us to see in a general way the model's performance.

3. RESULTS

During the investigation, experiments were ran with two different datasets. So, results will be presented by datasets.

The investigation experiments were essentially two:

1) Without Data Augmentation
2) With Data Augmentation

The first experiment was about running machine learning models on real data only. Dividing the dataset in training and test in portions of 60% and 40% respectively.

The second one was about, increasing real data by using data augmentation and use the synthetic data to train machine learning models and real data to validate them.

3.1 Dataset - 1

Table 1. Dataset 1 – Target Variable distribution

	Cure	**Fail**	**Total**
Target Variable instances	11	7	18

Experiment 1 results:

Table 2. Dataset 1 – Without Data Augmentation

Machine Learning Model	Precision	Recall	F1-Score
MLPClassifier	0.67	0.5	0.57
KNeighborsClassifier	1.0	0.25	0.4
LogisticRegression	0.0	0.0	0.0
RandomForestClassifier	0.0	0.0	0.0
SVC	0.0	0.0	0.0

Experiment 2 results:

Table 3. Dataset 1 – With Data Augmentation

Machine Learning - Data Augmentation Model	Precision	Recall	F1-Score
ADASYN			
MLPClassifier	0.88	1	**0.93**
LogisticRegression	0.78	1	0.88
KNeighborsClassifier	0.78	1	0.88
SVC	0.7	1	0.82
RandomForestClassifier	0.58	1	0.74
SMOTE			
SVC	0.88	1.0	**0.93**
MLPClassifier	0.88	1.0	0.93
LogisticRegression	0.78	1.0	0.88
KNeighborsClassifier	0.86	0.86	0.86
RandomForestClassifier	0.86	0.86	0.86
Borderline SMOTE			
KNeighborsClassifier	0.78	1.0	**0.88**
SVC	0.78	1.0	0.88
RandomForestClassifier	0.7	1.0	0.82
MLPClassifier	0.7	1.0	0.82
LogisticRegression	0.6	0.86	0.71
Gaussian Mixture Model			
LogisticRegression	0.5	0.57	**0.53**
KNeighborsClassifier	0.5	0.57	0.53
RandomForestClassifier	0.44	0.57	0.5
SVC	0.43	0.43	0.43
MLPClassifier	0.43	0.43	0.43
Genetic Algorithm			
LogisticRegression	0.75	0.86	**0.8**
KNeighborsClassifier	0.67	0.86	0.75
SVC	0.67	0.86	0.75
RandomForestClassifier	0.62	0.71	0.67
MLPClassifier	0.62	0.71	0.67
Generative Adversarial Network			
SVC	0.88	1.0	**0.93**
KNeighborsClassifier	1.0	0.71	0.83
LogisticRegression	0.64	1.0	0.78
RandomForestClassifier	0.71	0.71	0.71
MLPClassifier	0.71	0.71	0.71

3.2 Dataset – 2

Table 4. Dataset 2 – Target Variable distribution

	Cure	Fail	Total
Target Variable instances	189	58	247

Experiment 1 results:

Table 5. Dataset 2 – Without Data Augmentation

Machine Learning Model	Precision	Recall	F1-Score
MLPClassifier	0.67	0.24	0.35
KNeighborsClassifier	0.52	0.44	0.48
LogisticRegression	0.0	0.0	0.0
RandomForestClassifier	0.88	0.6	0.71
SVC	0.9	0.24	0.39

Experiment 2 results:

Table 6. Dataset 2 – With Data Augmentation

Machine Learning - Data Augmentation Model	Precision	Recall	F1-Score
ADASYN			
KNeighborsClassifier	0.93	0.98	**0.96**
RandomForestClassifier	0.81	0.83	0.82
MLPClassifier	0.49	0.78	0.6
LogisticRegression	0.36	0.62	0.46
SVC	0.29	0.55	0.38
SMOTE			
KNeighborsClassifier	0.86	0.98	**0.92**
RandomForestClassifier	0.73	0.91	0.81
MLPClassifier	0.69	0.91	0.79
SVC	0.4	0.72	0.52
LogisticRegression	0.38	0.6	0.46
Borderline SMOTE			
KNeighborsClassifier	0.81	0.95	**0.87**
RandomForestClassifier	0.75	0.78	0.76
MLPClassifier	0.51	0.83	0.63
LogisticRegression	0.33	0.71	0.45
SVC	0.3	0.88	0.45
Gaussian Mixture Model			
LogisticRegression	0.18	0.57	**0.28**
KNeighborsClassifier	0.18	0.57	0.28
RandomForestClassifier	0.18	0.57	0.28
SVC	0.18	0.57	0.28
MLPClassifier	0.18	0.57	0.28
Genetic Algorithm			
RandomForestClassifier	0.33	0.71	**0.45**
KNeighborsClassifier	0.32	0.72	0.44
MLPClassifier	0.26	0.84	0.4
SVC	0.27	0.72	0.39
LogisticRegression	0.26	0.83	0.39
Generative Adversarial Network			
KNeighborsClassifier	0.39	0.52	**0.45**
LogisticRegression	0.33	0.64	0.44
MLPClassifier	0.33	0.55	0.41
SVC	0.31	0.55	0.4
RandomForestClassifier	0.31	0.48	0.38

4. CONCLUSIONS

- The effectiveness of Glucantime as a treatment for Cutaneous Leishmaniasis was predicted using data augmentation and machine learning algorithms with F1-scores higher than 90%.

- The combination of data augmentation models and machine learning algorithms be- came very powerful given the problem conditions. The learning process was strength- ened by extracting significant characteristics from the original data to generate new data and training the machine learning models.

- In problems where data lack is a common factor, such as the medical field, generating more data with data augmentation techniques is highly useful. This is based on the improvement of the machine learning algorithms trained with augmented data compared to those trained only with real data, showing an improvement in the perfor- mance of up to 25%, which adds much value and can significantly impact the patient's health and life.

- Despite the difficulties that arose due to the lack of data to implement a prediction model, it was possible to follow a successful route that made it possible to take advan- tage of the data of each real patient. This route allowed to improve the predictions, as they were awful due to the little training that could be done without the data augmentation. It also helped to solve the impossibility of cross-validating because of minimum resulting partitions. For this reason, the route mentioned above was strategically developed so that the machine learning algorithms were trained only with synthetic data and were tested only with real data.

5. REFERENCES

[1] P. Zambrano, "Vigilancia y Análisis Del Riesgo En Salud Pública Protocolo De Vigilancia En Salud Pública Leishmaniasis," World health organization, vol. 02, p. 4, 2017.

[2] A. Masmoudi, N. Maalej, M. Mseddi, A. Souissi, H. Turki, S. Boudaya, S. Bouassida, and A. Zahaf, "Glucantime® par voie parentérale: Bénéfice versus toxicité," Medecine et Maladies Infectieuses, vol. 35, no. 1, pp. 42–45, 2005.

[3] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," no. 3, pp. 1322–1328, 2008.

[4] T. Ryan Hoens and N. V. Chawla, "Imbalanced datasets: From sampling to classifiers," Imbalanced Learning: Foundations, Algorithms, and Applications, pp. 43–59, 2013.

[5] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," Lecture Notes in Computer Science, vol. 3644, no. PART I, pp. 878–887, 2005

[6] S. Misra, H. Li, and J. He, Robust geomechanical characterization by analyzing the performance of shallow-learning regression methods using unsupervised clustering methods. Elsevier Inc., 2020.

[7] Y. Chen, M. Elliot, and J. Sakshaug, "A genetic algorithm approach to synthetic data production," ACM International Conference Proceeding Series, vol. 29-30-Augu, pp. 0–3, 2016.

[8] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F. Y. Wang, "Generative adversarial networks: Introduction and outlook," IEEE/CAA Journal of Automatica Sinica, vol. 4, no. 4, pp. 588–598, 2017

[9] S. Agarwal, Data mining: Data mining concepts and techniques. 2014.

[10] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random Forests and Decision Trees," International Journal of Computer Science Issues, vol. 9, no. 5, pp. 272–278, 2012.

[11] A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media.

[12] Tan Steinbach Kumar, Introduction to Data Mining (New International Editon). No. September, 2013.