Capturing Fairness and Uncertainty in Student Dropout Prediction – A Comparison Study

Efthyvoulos Drousiotis[1], Panagiotis Pentaliotis[1], Lei Shi[2], Alexandra I. Cristea[2]

[1] Department of Electrical Engineering & Electronics, University of Liverpool, Liverpool, UK
{e.drousiotis, p.pentaliotis}@liverpool.ac.uk
[2] Department of Computer Science, Durham University, Durham, UK
{lei.shi, alexandra.i.cristea}@durham.ac.uk

Abstract: This study aims to explore and improve ways of handling a continuous variable dataset, in order to predict student dropout in MOOCs, by implementing various models, including the ones most successful across various domains, such as recurrent neural network (RNN), and tree-based algorithms. Unlike existing studies, we arguably fairly compare each algorithm with the dataset that it can perform best with, thus 'like for like'. I.e., we use a time-series dataset 'as is' with algorithms suited for time-series, as well as a conversion of the time-series into a discrete-variables dataset, through feature engineering, with algorithms handling well discrete variables. We show that these much lighter discrete models outperform the time-series models. Our work additionally shows the importance of handing the uncertainty in the data, via these 'compressed' models.

Keywords: Discrete variables, Capturing Uncertainty, Time-Series, LSTM, BART, Prediction, MOOCs, Learning Analytics

1 Introduction

Over the years, an undeniable challenge in online learning became to find ways to reduce and predict students' dropout rates, which fall roughly at 77%-87% [3][4]. The majority of the studies such as [3][4], use the same dataset and variables to implement predictive models, without taking into consideration the type of variables each model uses for maximising its performance. For example, Tang *et. al.* [3] trained a time-series Long Short-Term Memory (LSTM) model using the same dataset that was used to train other non-time series machine learning models, including Logistic Regression, Random Forest, and Gradient Boosting Decision Tree (GBDT) models. The results show that time-series models (LSTM) outperform other machine learning models (i.e., Linear Regression, Decision Tree), and achieve higher accuracy, precision and recall when they are compared to their natural environment (continuous/time-series variables). We argue, however, that previous methods do not take into account the target on which the algorithms are performing best. We thus aim to provide benchmarks for predicting the completers and non-completers and examine the following research question:

Is it a good practice to use sequential time-series as-is, or first convert the dataset into a discrete-variables one, for obtaining enhanced metrics (precision, recall, accuracy) on predicting students' dropout with the appropriately tuned method?

2 Related Work

Many studies focused on classifying students into completers and non-completers. Some of them, such as [5], [6] use statistics, or traditional machine learning algorithms (e.g., Decision Trees, Logistic Regression, Random Forest, Support Vector Machines) [7]–[10], while others, such as [11], [12], used more advanced algorithms (e.g. Deep Learning), or even visualisation [13]. There are also a few studies [3], [4], that used both traditional machine learning algorithms and more advanced. However, they [3], [4] used the same dataset to train both Neural Networks and machine learning models (time-series), showing that NN outperformed the other machine learning techniques. In our case, we convert the time-series dataset through feature engineering into discrete variables and train each model on the type of dataset it can process best. For example, [14] indicates that if our aim were to train a Neural Network, it is better to use a time-series dataset, while [15] suggests that we should use discrete variables when we aim to train a tree-based algorithm (either categorical or continuous variables).

Interestingly, some papers [12] [13] show that Artificial Recurrent Neural Networks (RNN) with memory, such as Long-Short-Term-Memory (LSTM), are generally considered as superior models to solve time-series tasks, because of their nature – the way they operate and handle data. On the other hand, [18], [19] indicate that traditional machine learning algorithms, such as Logistic Regression, Random Forest and GBDT produce better results with discrete-variable data.

3 Method

The dataset used in this study is comprising 300,000 interactions and 2,000 unique registered students, extracted from XuetangX (launched in October 2013, one of the largest MOOC platforms in China). We converted the time-series dataset, which our LSTM model was trained on, into a discrete-variables dataset, which our tree-based models were trained on. For the construction of the discrete-variables dataset, we used the time-series dataset and we have counted for each student the number of unique actions. In total, there are 14 different types of unique actions and thus we engineered 14 features for 14 input variables for our predictive models. Considering the LSTM model's feature engineering in preparation of the dataset, the actions of each student were sequentially grouped together, according to the time they were performed. Thus, the essence of the time-series was preserved, while still considering the unique action performed. Afterwards, the actions were translated into a sequence of binary numbers, to retain the categorical nature of the actions. Here, we examined the effectiveness of converting a time-series dataset into a discrete dataset through feature engineering. We trained the predictive models with the initial raw datasets, aiming to produce a benchmark for future work. We implemented an LSTM model and several tree-based machine learning models, including Decision Tree, Random Forest, and BART.

For all the above models we used the basic parameters, including the basic split of the data into 70% train and 30% test sets. Moreover, to evaluate the machine learning models, we used the k-fold cross-validation technique, and we did not perform any hyperparameter optimisation. The purpose of this setting is to find a benchmark and compare the two datasets on their primitive forms, without any data pre-processing (sequential time-series and discrete). To evaluate our predictive model's performance, we utilised the following standard, comprehensive metrics:

- *Precision*: the proportion of positive identifications which was actually correct;
- *Recall*: the proportion of actual positives that were identified correctly;
- *F1 score*: the weighted average of Precision and Recall;
- *Accuracy*: the ratio of correctly predicted observations over the total observations.

4 Results and Discussions

Table 1 presents the result comprising three tree-based models (Decision Tree, Random Forest, BART) and an LSTM model. From the results, we can clearly determine the difference between the two types of datasets and draw some useful conclusions. BART outperforms the other models – achieving a very high accuracy of 90% for identifying students who might drop out from an online course. The Decision Tree and Random Forest models achieved relatively high accuracy of 83% and 89%, respectively. The LSTM model achieved the lowest accuracy of 77%. Table 1 especially showcases the performance of the BART model and its improved learning ability in comparison with the other models. From the four figures (Figs. 1-4), and the AUC scores, we observe that BART (Fig. 3) has an improved ability to discriminate the test values in comparison with the other models (Decision Tree, Random Forest, LSTM). Furthermore, we can identify the improved trained ability of the tree-based models, when the discrete dataset was used, by the recall metric, which shows a clear ability to select the most relevant items on the classification task with the highest percentage of 96% produced by BART. In comparison with the tree-based models, the LSTM model did not perform as well as the other models. That is partially because LSTMs are known to require a large amount of data, in order to be efficiently trained.

Table 1. Performance comparisons between the predictive models

Metric	DT	RF	BART	LSTM
Precision	0.83	0.88	**0.89**	0.77
Recall	0.83	0.88	**0.96**	0.76
F1	0.83	0.87	**0.92**	0.75
Accuracy	0.82	0.89	**0.90**	0.77

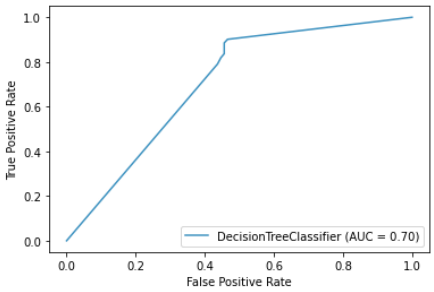

Fig. 1 Decision Tree ROC curve

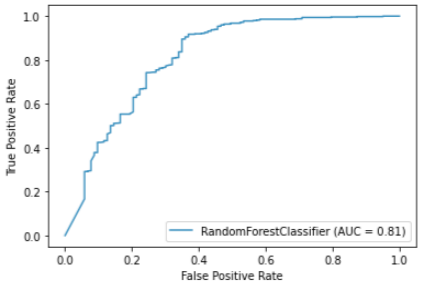

Fig. 2 Random Forest ROC curve

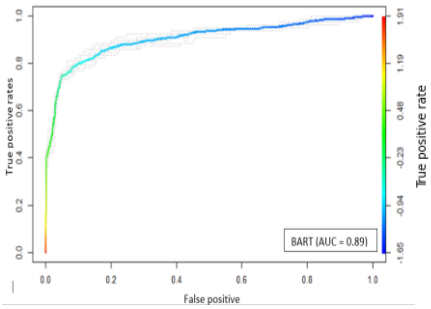

Fig. 3 BART ROC curve

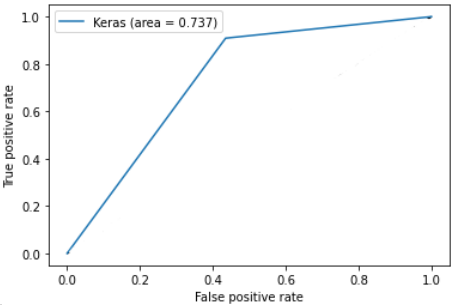

Fig. 4 LSTM ROC curve

Our results suggest that, whenever possible, it could be beneficial to convert the time-series dataset into a discrete variable dataset, as it is highly probable to produce better performance, especially when the time-series datasets are not populated enough.

5 Conclusions

In summary, this paper presents the results of a study aiming to discover whether it is efficient to convert a time-series dataset into discrete variables dataset, to train predictive models with better performance, in terms of predicting students' dropout. The research results have clearly indicated that we should convert a dataset into different forms when this is feasible. It has shown that this process assists different types of predictive models to obtain higher performance and enhance their learning ability. We have proven that it would be useful to manipulate the dataset for a variety of models first, thus enhancing the final results. We have also shown that BART, which includes a representation of uncertainty, outperforms all other tree-based methods.

Future work might include tuning the models' parameters and investigating the dataset further through data pre-processing and more sophisticated feature engineering techniques (i.e., Frequency count, Frequency Encoding) to achieve better performance. Also, it would be interesting to perform hyperparameter optimisation so that we can find out the optimal learning efficiency of the predictive models. In addition to improving the algorithms, more data could refine the results of this study.

References

[1] C. Gütl, R. H. Rizzardini, V. Chang, and M. Morales, 'Attrition in MOOC: Lessons Learned from Drop-Out Students', in *Learning Technology for Education in Cloud. MOOC and Big Data*, Cham, 2014, pp. 37–48, doi: 10.1007/978-3-319-10671-7_4.

[2] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart, 'Predicting MOOC Dropout over Weeks Using Machine Learning Methods', in *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, Doha, Qatar, Oct. 2014, pp. 60–65, doi: 10.3115/v1/W14-4111.

[3] C. Tang, Y. Ouyang, W. Rong, J. Zhang, and Z. Xiong, 'Time Series Model for Predicting Dropout in Massive Open Online Courses', in *Artificial Intelligence in Education*, Cham, 2018, pp. 353–357, doi: 10.1007/978-3-319-93846-2_66.

[4] L. Wang and H. Wang, 'Learning Behavior Analysis and Dropout Rate Prediction Based on MOOCs Data', in *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, Aug. 2019, pp. 419–423, doi: 10.1109/ITME.2019.00100.

[5] A. Cristea, A. Alamri, C. Stewart, M. Alshehri, and L. Shi, 'Earliest Predictor of Dropout in MOOCs: A Longitudinal Study of FutureLearn Courses Mizue Kayama', presented at the *27th International Conference on Information Systems Development (Isd2018 Lund, Sweden)*, Aug. 2018.

[6] M. Zhu, Y. Bergner, Y. Zhang, R. Baker, Y. Wang, and L. Paquette, 'Longitudinal engagement, performance, and social connectivity: a MOOC case study using exponential random graph models', in *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge - LAK '16*, Edinburgh, United Kingdom, 2016, pp. 223–230, doi: 10.1145/2883851.2883934.

[7] A. Alamri *et al.*, 'Predicting MOOCs Dropout Using Only Two Easily Obtainable Features from the First Week's Activities', in *Intelligent Tutoring Systems*, vol. 11528, A. Coy, Y. Hayashi, and M. Chang, Eds. Cham: Springer International Publishing, 2019, pp. 163–173.

[8] J. Chen, J. Feng, X. Sun, N. Wu, Z. Yang, and S. Chen, 'MOOC Dropout Prediction Using a Hybrid Algorithm Based on Decision Tree and Extreme Learning Machine', *Mathematical Problems in Engineering*, Mar. 18, 2019. https://www.hindawi.com/journals/mpe/2019/8404653/ (accessed Feb. 02, 2021).

[9] C. Jin, 'MOOC student dropout prediction model based on learning behavior features and parameter optimization', *Interactive Learning Environments*, pp. 1–19, Aug. 2020, doi: 10.1080/10494820.2020.1802300.

[10] F. D. Pereira *et al.*, 'Early Dropout Prediction for Programming Courses Supported by Online Judges', in *Artificial Intelligence in Education*, Cham, 2019, pp. 67–72, doi: 10.1007/978-3-030-23207-8_13.

[11] M. Fei and D. Yeung, 'Temporal Models for Predicting Student Dropout in Massive Open Online Courses', in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, Nov. 2015, pp. 256–263, doi: 10.1109/ICDMW.2015.174.

[12] J. Gardner and Y. Yang, 'Modeling and Experimental Design for MOOC Dropout Prediction: A Replication Perspective', *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, p. 10, 2019.

[13] A. Alamri, Z. Sun, A. I. Cristea, G. Senthilnathan, L. Shi, and C. Stewart, 'Is MOOC Learning Different for Dropouts? A Visually-Driven, Multi-granularity Explanatory ML Approach', in *Intelligent Tutoring Systems*, Cham, 2020, pp. 353–363, doi: 10.1007/978-3-030-49663-0_42.

[14] 'Time series forecasting | TensorFlow Core', *TensorFlow*. https://www.tensorflow.org/tutorials/structured_data/time_series (accessed Feb. 10, 2021).

[15] ‘Decision Tree - Overview, Decision Types, Applications’, *Corporate Finance Institute*. https://corporatefinanceinstitute.com/resources/knowledge/other/decision-tree/ (accessed Feb. 10, 2021).
[16] F. A. Gers, D. Eck, and J. Schmidhuber, ‘Applying LSTM to Time Series Predictable Through Time-Window Approaches’, in *Neural Nets WIRN Vietri-01*, London, 2002, pp. 193–200, doi: 10.1007/978-1-4471-0219-9_20.
[17] X. Zhang, X. Liang, A. Zhiyuli, S. Zhang, R. Xu, and B. Wu, ‘AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction’, *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 569, p. 052037, Aug. 2019, doi: 10.1088/1757-899X/569/5/052037.
[18] I. K. Sethi and B. Chatterjee, ‘Efficient decision tree design for discrete variable pattern recognition problems’, *Pattern Recognition*, vol. 9, no. 4, pp. 197–206, Jan. 1977, doi: 10.1016/0031-3203(77)90004-8.
[19] Y. SONG and Y. LU, ‘Decision tree methods: applications for classification and prediction’, *Shanghai Arch Psychiatry*, vol. 27, no. 2, pp. 130–135, Apr. 2015, doi: 10.11919/j.issn.1002-0829.215044.