EARLY EDUCATIONAL PERFORMANCE PREDICTION A DEEP LEARNING APPROACH

DỰ BÁO KẾT QUẢ HỌC TẬP CỦA SINH VIÊN BẰNG PHƯƠNG PHÁP HỌC SÂU

Nguyen Dinh Van[1,*], Nguyen Viet Tung[1], Ha Van Phuong[2]

DOI: https://doi.org/10.57001/huih5804.65

ABSTRACT

Early performance prediction is crucial for educators to identify struggling students. This is especially important in a university where good students can perform badly due to multiple external challenges. However, there are huge differences in terms of programs, policies as well as culture between universities. These differences contribute significantly to students' academic performance. Thus, it is important to address different universities separately to predict students' performance accurately. In this paper, an analysis of nearly 400 students' records across 7 semesters of the same major in Hanoi University of Science and Technology is presented. Because of the university privacy policy, it is impossible to obtain students information other than their academic results. In addition, due to the modest size of the datasets, imbalanced data is expected. Hence, we propose to use the Borderline SMOTE algorithm to reduce the dataset's imbalanced distribution. The data is then fed into a deep neural network to predict students' performance of the 4th year based on their previous years' scores. A promising result of 77% accuracy is achieved.

Keywords: *Education, machine learning, performance prediction, data analysis.*

TÓM TẮT

Dự đoán kết quả sớm là rất quan trọng đối với các nhà giáo dục để xác định những học sinh đang gặp khó khăn. Điều này đặc biệt quan trọng trong một trường đại học nơi sinh viên giỏi có thể có thành tích kém do nhiều thách thức bên ngoài. Tuy nhiên, có sự khác biệt rất lớn về chương trình, chính sách cũng như văn hóa giữa các trường đại học. Những khác biệt này góp phần đáng kể vào kết quả học tập của học sinh. Do đó, để dự đoán chính xác kết quả học tập của sinh viên, việc thực hiện nghiên cứu cho từng trường đại học là cần thiết. Trong bài báo này, nhóm tác giả đã phân tích hồ sơ của gần 400 sinh viên trong 7 học kỳ của cùng một chuyên ngành tại Trường Đại học Bách khoa Hà Nội. Vì chính sách bảo mật thông tin của nhà trường, nhóm tác giả chỉ nhận được các thông tin về kết quả học tập của sinh viên. Ngoài ra, do kích thước bộ dữ liệu còn khiêm tốn, sự mất cân bằng trong dữ liệu là hoàn toàn có thể xảy ra. Do đó, chúng tôi đề xuất sử dụng thuật toán Borderline SMOTE để giảm sự mất cân bằng của tập dữ liệu. Sau đó, dữ liệu được đưa vào một mạng nơ-ron học sâu để dự đoán kết quả học tập của học sinh trong năm thứ 4 dựa trên điểm số của các năm trước đó. Kết quả thu về cho thấy mạng học sâu có thể dự báo chính xác kết quả học tập năm thứ 4 đến 77%.

Từ khóa: *Giáo dục, học máy, dự báo kết quả học tập, phân tích dữ liệu.*

[1]SEEE, Hanoi University of Science and Technology
[2]Hanoi University of Industry
[*]Email: van.nguyendinh@hust.edu.vn
Received: 25/9/2022
Revised: 18/11/2022
Accepted: 22/11/2022

1. INTRODUCTION

Almost every university nowadays is using a database management system to store students scores. As the data getting bigger, a question a rise of whether this information can be used to learn the insight about the academic performance of both students and lecturers. This is where educational data mining (EDM) [1] takes place. There are different approaches in EDM given the widely different in educational system and culture between countries, cities and even major within a university. However, the ultimate goal of EDM is simple: to enhance the educational outcome.

Among different subjects in EDM, students' performance prediction is often considered to be the most important one [2]. If the early prediction of failing students can be done, universities can provide much needed support and help for these individuals.

To predict students' academic performance, three types of data are used:

Type 1: Personal information (e.g., age, gender, hometown, etc.)

Type 2: Personal interaction with courses (e.g. number of attempts for a homework, number of absences, class interactions, etc.)

Type 3: Scores (midterm, final, average scores for classes)

However, more often, both type 1 and type 2 data are not readily available due to privacy concerns. Hence, students' scores remain the single most important factor to evaluate and predict students' progress. Another challenge for educational performance prediction is that every educational system is complex and unique. Hence, to make a prediction, it often requires a deep understanding of the system.

In this paper, a dataset of 356 students across 7 semesters (3 and a half years) is

described and analyzed. Since the dataset size is modest, imbalanced data distribution is expected. We apply a version of Borderline SMOTE algorithm to enhance data distribution without changing the dataset characteristics. Using a proposed deep neural network structure, we can predict the 7th semester performance of students using his/her previous scores with 77 percent accuracy. We also compare multiple machine learning techniques to find out the optimal algorithm for this EDM problem.

2. STATE OF THE ART

In EDM, various approaches are proposed such as: Classification and Regression, Clustering, Association Rule Mining, Discovery with Models, Outlier Detection, Sequential Pattern Mining or Visualization Techniques, etc. [3, 5-8]. However, the three most frequently used techniques are: Classification and Regression, Clustering and Association Rule Mining [9].

Classification is a supervised learning technique that allows grouping students into known categories. This is extremely effective in the setting of a university since students Grade Point Average (GPA) are already divided into several known classes. Some well-known works in this direction are [10, 11].

Regression is often used for precise score prediction as in [12] where authors attempt to predict intermediate and secondary students with a RMSE of 5.34 for scores in scale of 100.

Other works can be found [13] presented a bigdata based recommender system. The work uses association rules mining, which is an unsupervised method, to find the potential relationship between student academic activities. Big data tools such as Spark and Hadoop are used to facilitate the system. The obtained results show the top rules efficiency are around 95 to 98 percent with confidence are between 0.69 to 1.0.

In general, regression methods only work with big datasets with at least 2 types of data. This is because there are multiple factors affecting the precise score of a student for a course. Some of these factors are often impossible to quantify or be recorded. This leads to a more popular classification method which can approximately find the correct category for each student in terms of performance. In this paper, due to the limited size of the dataset as well as only type 3 data is available, a classification method is chosen.

3. METHODOLOGY

3.1. Pre-processing and analysis

The dataset is collected from 2016 to 2019 for 429 students of the same major from a university in Vietnam. However, only 356 students have more than 7 semesters. Hence, to predict the 7th semester performance, we filtered out all students who has less than 7 semesters of information.

The datasets have 22625 records, each is records represents a student's score for one class. The column information is shown as in Table 1.

Table 1. Column description of the raw dataset

Name	Type	Description
StudentID	Int	Encoded value for students' id (from 1 - 430)
Semester	Int	The semester from 2016 to 2019
CourseID	Int	Course identification number
CourseName	String	Name of the course
Credit	Int	Number of credits
ClassID	Int	Class identification number for the class
Midterm	Float	Student's midterm score
Final	Float	Student's final score
LetterScore	String	Letter score of the student

The letter score is converted from scale of 10 score as in Table 2.

Table 2. Letter score to numerical score conversion

Letter	Value	Letter	Value
A+	9.5 - 10	C	5.5 – 6.4
A	8.5 - 9.4	D+	5.0 – 5.4
B+	8.0 - 8.4	D	4.0 – 4.9
B	7.0 - 7.9	F	0.0 – 3.9
C+	6.5 - 6.9		

We also perform 4 steps of data pre-processing:

Step1: calculate the average score based on the midterm and the final score using (1) (provided by the university)

$$\text{Final\_avg} = \text{Midterm}*0.3 + \text{Final}*0.7 \quad (1)$$

Since the range for each letter score vary, taking only the letter score as the average score for a student in class would cause loss of information. Thus, taking the final average score can potentially improve the prediction outcome.

Step 2: For the same academic year, if a student takes the same class more than once, the maximum final average score will be selected. This process eliminates the data duplication that happens when students retake courses. After step 2, every student will only have one final average score for a particular class.

Step 3: Calculate the academic year's final average score using both the course final average score and the number of credits assigned for each course. The calculation is done as in (2). Since each course has a different number of credits, considering the number of credits will act as a standardization method for the academic year average score.

$$\text{avrScore} = \frac{\sum(\text{course\_credit}*\text{final\_avg})}{\sum \text{credits}} \quad (2)$$

Step 4: from the initial raw dataset, create a new dataset for all 356 students. The new dataset has 356 rows with

each row corresponding to a student record and 9 columns described in Table 3. Since the major of all students in this dataset is in natural science, only major courses and other natural science courses (i.e. Math, Programming, Physics, etc.) will be considered. Other supplementary courses and non-credit courses will be discarded. The pre-processed dataset is described in Table 3.

Table 3. Columns description for the pre-processed dataset

Name	Type	Description
EE1	Float	First year major courses average score
EE2	Float	Second year major courses average score
EE3	Float	Thrid year major courses average score
EE4	Float	Fourth year major courses average score
MI	Float	Math courses average score
IT	Float	Programming courses average score
PH	Float	Physics courses average score
ME	Float	Mechanical engineering average score

Step 5: Convert EE4 average score into categorical data for classification problem using the following rules (3):

$$\text{at\_score} = \begin{cases} 0 & \text{score} \in [0,5) \\ 1 & \text{score} \in [5,6) \\ 2 & \text{score} \in [6,7) \\ 3 & \text{score} \in [7,8.5) \\ 4 & \text{score} \in [8.5,10) \end{cases} \quad (3)$$

After this step, we form a classification problem: given a set of average score EE1, EE2, EE3, MI, IT, PH, ME, the student performance in the 4th year (represented by EE4) will be predicted.

3.2. Data distribution analysis

One problem usually found in EDM is the imbalanced distribution of data. This happens because more students have average scores than the two extreme ends (low or high score). The distribution is expected to closely follow Gaussian distribution if the number of students is large enough. However, since our goal is to predict if a student is going to perform badly (i.e., receive low scores), this creates a problem for any classification algorithm.

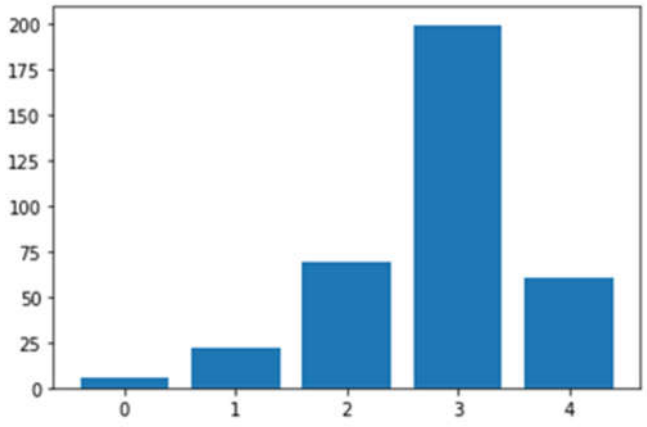

Figure 1. Original classes distribution for EE4 in the pre-processed dataset

For the pre-processed dataset, the distribution of EE4 classes is presented in Fig. 1. This figure shows a huge difference between the number of the low score classes (class 0, 1), the middle score classes (class 2, 3) and high score classes (class 4).

To reduce the impact of classes misbalancing, we propose to use Borderline SMOTE oversampling algorithm [14] to introduce more data for both end classes. Since the algorithm only introduces data for edge of each class, it is expected to keep the characteristic of the entire dataset remain the same. After sampling, 995 rows of record are received with each class (0 - 4) having equally 199 records. The new classes distribution is shown in Fig. 2.

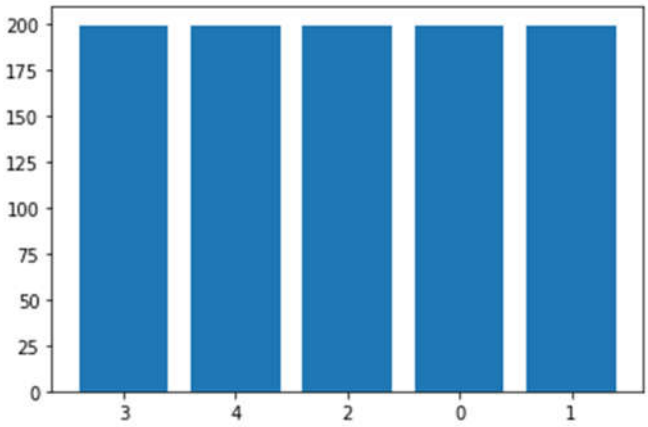

Figure 2. Oversampling dataset distribution

3.3. Deep Neural Network

With 995 records, each has 7 features and 5 different classes to predict, numerous machine learning techniques will be studied and compared. To facilitate the process of training and validation, we split the dataset into 80% for training and 20% for test. Within the training dataset, we also use 10% for model validation.

Among different machine learning techniques, the best method is a 7 layers deep neural network. The network structure will be presented as follows:

- The input layer: a 7 nodes input layer for 7 average score features respectively: EE1, EE2, EE3, MI, IT, PH, ME

- Each of the 5 hidden layers each has 32, 64, 32, 16, 8 nodes respectively with the chosen activation function to be ReLU.

- The output layer: 5 nodes corresponding to 5 classes for EE4 using softmax activation function.

- The Root mean square propagation (RMSprop) is used to propagate error with categorical cross entropy loss function. Batch size is set to 15 with 200 epochs for training.

- The details for model parameters selection are presented in the next section.

3.4. Parameters Selection

To find the optimal parameters, set for this deep neural network, various experiments are performed. Firstly, a different number of epochs are tested. The result is shown

in Table 4. With different numbers of epochs, after 200 epochs, the model accuracy and loss are at the optimal point of 0.69 loss and 0.76 accuracy.

In addition, different optimizers are also surveyed at 200 epochs. The result is shown in Table 5. there are 4 optimizers tested: Adaptive moment (Adam), Root mean squared propagation (RMSProp), Adadelta, stochastic gradient descent (SGD). The RMSProp achieved the best accuracy with the lowest loss of 0.76 and 0.69 respectively.

Table 4. Training result with different number of epochs

Number of Epochs	Test Loss	Test Accuracy (%)
50	0.8716	67.12
100	0.8378	67.12
200	0.6925	76.71
500	1.5327	76.03

Table 5. Model accuracy using different optimizers

Optimizer	Test Loss	Test Accuracy (%)	Optimizer
Adam	0.8583	69.18	Adam
RMSProp	0.6925	76.71	RMSProp
Adadelta	1.7323	24.66	Adadelta
SGD	0.8579	69.86	SGD

4. RESULTS AND DISCUSSION

4.1. Deep Neural Network Result

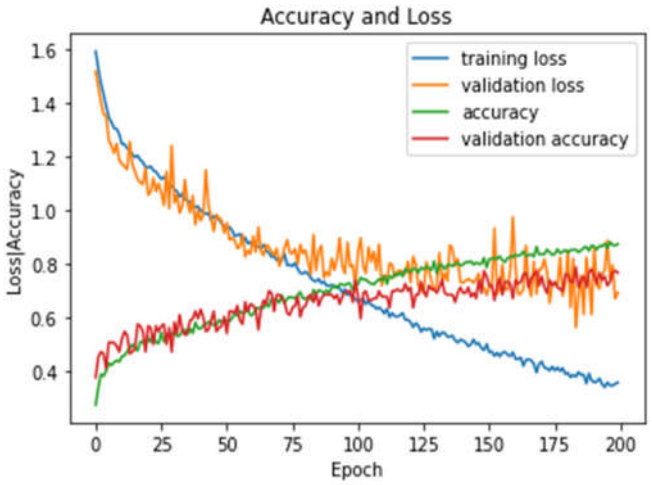

Figure 3. Model accuracy and losses after 200 epochs

The training process is carried out with 200 epochs. Results of training is shown in Fig. 3. The best accuracy on training set is achieved at 200 epochs at 85% while it reaches 74% and 76% of accuracy on validation and test set respectively.

Since the dataset is quite small, a k-fold validation is performed with k = 10. The accuracy of the model stays at 75% (± 4%) with categorical cross-entropy loss around 0.78. This demonstrates the reliability of the proposed model.

4.2. Model comparison

In EDM, each dataset carries different characteristics and represents different academic systems. Thus, it is quite difficult to have a direct comparison between proposed solutions. Given that problem, we will try to compare different machine learning techniques with the proposed deep neural network model. Popular techniques in EMD such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), and Naïve Bayesian (NB) are studied. The configuration for each technique is shown in Table 6.

Table 6. Model Configuration

Model	Configuration
KNN	n_neighbors = 5
SVM	kernel = 'linear', C = 1, probability = True
RF	n_estimators = 20, max_depth = 5
DT	max_depth = 5
NB	Default

With the above configurations, we perform training and validation on the same dataset. The accuracy of each model is shown in Fig. 4.

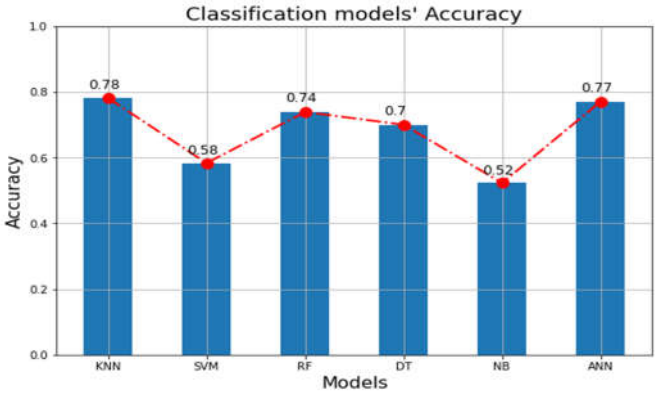

Figure 4. Classification models accuracy

Given the results showed above, KNN, RF and the proposed deep neural network (ANN) has a similar accuracy of 0.75 ± 0.03. The accuracy of SVM and NB is relatively low.

To further investigate the accuracy of models, we proceed to calculate losses based on log loss function in (4).

$$L_{\log(y,p)} = -(y\log(p) + (1-y)\log(1-p)) \quad (4)$$

With y is the true class, p is predicted class.

The log loss for each model is presented in the Fig. 5.

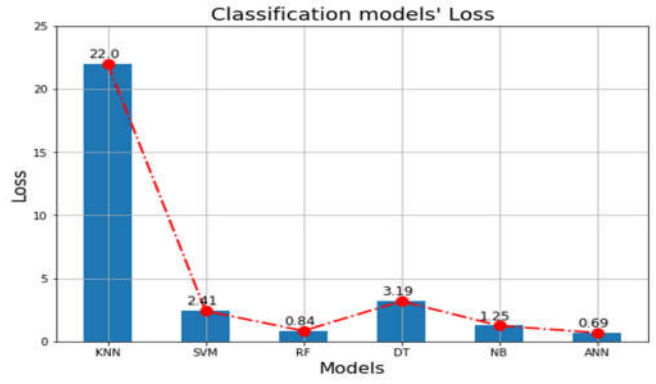

Figure 5. Models Log Loss comparision

It can be observed that although KNN achieves a high accuracy, the loss is much higher than other models. This can lead to a sharp decrease in accuracy when new data is introduced. Other

5. CONCLUSION

In this paper, a deep neural network model is proposed to predict students' performance. From the initial imbalanced dataset of 356 students (exclude ones with incomplete or incorrect data, etc.), we porposed to use Borderline SMOTE over-sampling method to enhance the distribution of classes. We also make an assumption that the student performance will mostly depend on major courses as well as natural science courses (i.e. Math, Information Technology, Physic, Mechanical Engineering, et.).

Having a evenly distributed dataset of 995 records, a deep neural network of 7 layers is designed to predict the students performance in the 7th semester (4th year). the result is promising with accuracy of 77%. This enables the university anticipate early sign of warning students.

We also compare the proposed ANN with different machine learning techniques such as KNN, SVM, RF, etc. to find the optimal one. Although KNN achieves the best accuracy among all models, its loss is way too high thus it is expected to perform badly with new data. Hence, proposed ANN remains the best model for the particular dataset.

In the future, with a bigger dataset with potentially data of type 1 and type 2, it is expected the prediction accuracy will be increased.

REFERENCES

[1]. S. H. Ganesh, A. J. Christy, 2015. *Applications of educational data mining: a survey*. in 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE, pp. 1–6

[2]. Albreiki B, Zaki N, Alashwal H., 2021. *A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques.* in Education Sciences, 11(9):552

[3]. C. Romero, S. Ventura, 2010. *Educational data mining: a review of the state of the art*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 6, pp. 601–618.

[4]. A. Pena-Ayala, 2014. *Educational data mining: A survey and a data mining-based analysis of recent works.* Expert systems with applications, vol. 41, no. 4, pp. 1432–1462.

[5]. B. Bakhshinategh, O. R. Zaiane, S. ElAtia, D. Ipperciel, 2018. *Educational data mining applications and tasks: A survey of the last 10 years.* Education and Information Technologies, vol. 23, no. 1, pp. 537–553.

[6]. S. Agarwal, G. Pandey, M. Tiwari, 2012. *Data mining in education: data classification and decision tree approach*. International Journal of eEducation, e-Business, e-Management and e-Learning, vol. 2, no. 2, p.140.

[7]. P. D. Antonenko, S. Toy, D. S. Niederhauser, 2012. *Using cluster analysis for data mining in educational technology research.* Educational Technology Research and Development, vol. 60, no. 3, pp. 383–398.

[8]. S. Hussain, 2017. *Survey on current trends and techniques of data mining research*. London Journal of Research in Computer Science and Technology, vol. 17, no. 1, pp. 7–16.

[9]. C. Jalota, R. Agrawal, 2019. *Analysis of Educational Data Mining using Classification.* 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 2019, pp. 243-247, doi: 10.1109/COMITCon.2019.8862214.

[10]. M. Á. Prada, et al., 2020. *Educational Data Mining for Tutoring Support in Higher Education: A Web-Based Tool Case Study in Engineering Degrees*. in IEEE Access, vol. 8, pp. 212818-212836, doi: 10.1109/ACCESS.2020.3040858.

[11]. Yousafzai B.K., Hayat M., Afzal S. 2020. *Application of machine learning and data mining in predicting the performance of intermediate and secondary education level student*. Education and Information Technology 25, 4677–4697, https://doi.org/10.1007/s10639-020-10189-1

[12]. Dahdouh K., Dakkak A., Oughdir L., et al. 2019. *Large-scale e-learning recommender system based on Spark and Hadoop.* J Big Data 6, 2. https://doi.org/10.1186/s40537-019-0169-4

[13]. Han H., Wang WY., Mao BH., 2005. *Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning.* Advances in Intelligent Computing. ICIC 2005. Lecture Notes in Computer Science, vol 3644. Springer.

THÔNG TIN TÁC GIẢ

Nguyễn Đình Văn[1], Nguyễn Việt Tùng[1], Hà Văn Phương[2]

[1]Trường Điện - Điện tử, Trường Đại học Bách khoa Hà Nội

[2]Khoa Điện, Trường Đại học Công nghiệp Hà Nội