Predicting Student Performance by Using Neural Network

Qingli Guo
Email: u6011074@anu.edu.au

Research School of Computer Science, Australian National University

Abstract. With the expansion of education database, humans are hard to estimate all students' performance. It is necessary to develop a tool which can automatically predict student performance. This paper implements a neural network classification model to predict whether the student will pass or fail in the mathematic course. Besides, autoencoder method and genetic algorithm are applied to improve the neural network model. Compared to the Paulo and Alice's published research paper for the same UCI dataset, our basic model can get better performance which can achieve 92% accuracy, and their model can achieve around 89% accuracy. After applying AutoEncoder method and genetic algorithm, our model can achieve the highest 96% accuracy.

Keywords: Student Performance, Classification, Neural Network, AutoEncoder, Genetic Algorithm, Feature Selection

1 Introduction

Predicting student performance is a useful application for school, educator and students. School can admit high quality students according students' academic performance. For the educator, it can help them monitor students' performance and provide better education methods. For students, they can improve and change performance in time.

There are many studies have discussed this similar topic. Amirah et al. [1] found that the cumulative grade point average(CGPA) was an important attribute and had been frequently been used. They explored several prediction models, including Decision Tree, Neural Network, Naïve Bayes, K-nearest and Support Vector Machine. The result was that Neural Network can have the highest accuracy, and the following is Decision Tree. Besides, nowadays, there is a lot of distance education, which students often feel isolated due to lack of communication. It is important for tutors to recognize students, who have high probability with bad performance, and take precautions [2]. Kotsiantis et al. proposed to use ensemble classifiers, which combined an incremental version of the WINNOW, the 1-NN and the Naïve Bayes algorithms to improve the accuracy for predicting students' performance in distance education [2].

This paper uses the real-world dataset from two Portuguese secondary schools, which was collected by school reports and questionnaires [3]. The dataset includes 33 attributes, such as student past grades, social, family, school related features. The detailed attributes are shown in Appendix Table 7 attribute, description and domain columns. Mathematics is an important course for students. This mathematics dataset has 395 instances. The objective is to predict whether students can pass or fail the mathematics course in the final year. We use the neural network model to solve this binary classification problem (pass or fail). Besides, we apply autoencoder method and genetic algorithm to improve the model and compare the result with published paper with the same dataset by using accuracy and f-score metrics. The final part of the paper describes conclusion and future work.

2 Method

This part will discuss how to preprocess the data, implement the neural network, implement the evaluation method and improve the neural network. The development framework is PyTorch.

2.1 Preprocess the Data

In the dataset, each row represents a student information. We need to read the original dataset file, split every line data, remove quote and get every attribute. The original input and output values are string type. For convenient use, we need to change them to numeric values. For example, values of school attribute can be "GP" and "MS", and we can set "GP" to class 0 and "MS" to class 1. If the age is the string numeric, we can just change it to the integer value. The detailed transformation for every attribute is shown in Appendix Table 7 value and class columns. The dataset has 33 attributes. We split 80% data for training and 20% data for testing. Also, we need to split input features and target output. The previous 32 attributes are input features and the last one G3 attribute is the target output. The neural network implementation is using PyTorch, so we need to create Tensors and convert them to Variables, due to the reason that the PyTorch only trains the neural network in Variables.

2.2 Implement the Neural Network

The basic neural network has three layers, including input layer, hidden layer and output layer. Each layer has different number of neurons. There are 32 attributes for input features, so the input layer has 32 neurons. For the hidden layer, the number of neurons is 50. The last attribute G3 score is the output, which has two possibilities (pass or fail), so the output layer neurons are 2. Fewer epochs often can't get good result, in our neural network model, we set number of epoch is 3000. Also, learning rate is an important hyper parameter in the neural network model. If the learning rate is high, the training process may not converge and miss the local minimal point due to the big step [9]. If the learning rate is low, the step is very small and the optimization will cost a lot of time to converge [9]. We experimented different learning rate and set it to 0.01. Next, we apply cross entropy loss function and SGD optimizer. We can train our neural network by using training dataset, compute loss, clear the gradients and perform backward pass. Then, using test data to test the model by evaluating accuracy and f score. Besides, we can try different parameters to get better result, such as learning rate, number of hidden layer, activation function, etc.

2.3 Implement the Evaluation Method

The efficiency of the machine learning models can be determined by using true positive rate(TP), false positive rate(FP), true negative rate(TN), and false negative rate(FN) [4].
a) TP represents the number of instances that are predicted positive and actually are positive.
b) FP represents the number of instances that are predicted positive and actually are negative.
c) TN represents the number of instances that are predicted negative and actually are negative.
d) FN represents the number of instances that are predicted negative and actually are positive.

In this paper, we use the accuracy metric to evaluate the model performance. Accuracy is the percentage of correctly classified result, which is a very commonly used and easily to calculate [5].

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN) \quad (1)$$

Besides, we can use confusion matrix [4] to calculate the TP, TN, FP, FN.

Table 1. Confusion Matrix

Actual	Predicted	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

However, sometimes accuracy is not enough to measure the model performance [6]. For example, if now we have 100 students' data, 99 students passed the exam and 1 failed the exam. Besides, pass means 0 and fail means 1. Our model predicted all passed 99 students, but failed to detect the student who failed the exam, which is more important for us.

$$TN = 99, TP = 0 \quad (2)$$

$$\text{Accuracy} = (99+0) / 100 = 99\% \quad (3)$$

Now, the accuracy is 99%, but is not good to process unbalanced data. Thus, we can also use another measurement, which is also widely used, such as precision, recall, f-score [7].

Precision:

$$\text{Precision} = TP / (TP+FP) \quad (4)$$

Recall:

$$\text{Recall} = TP / (TP+FN) \quad (5)$$

F-Score:

$$F = (2*\text{Precision}*\text{Recall}) / (\text{Precision}+\text{Recall}) \quad (6)$$

2.4 AutoEncoder

Many methods have been proposed to improve the neural network, and we are inspired by the idea of image compression which was raised by Gedeon and Harris [8]. By using the compression method, it can remove some redundant units and make all units become more meaningful [8]. Duplicated units will increase the size and decrease the speed. We apply this idea and implement it by using PyTorch autoencoder class.

For high dimensional data, noise features will increase the space and computation time. Some features maybe duplicated and correlated to other features, so it's necessary to select most important features and reduce the data dimensions [10]. AutoEncoder technology can reconstruct the input data, produce similar output as input, and get more discriminative unites, which can seek the compressed dataset interpretation and preserve the most essential information [10].

We create the AutoEncoder class and implement to encode input data. The process of using AutoEncoder to implement neural network model is similar to before process. The difference is that we mentioned before that our input layer neurons are 32, and now we use AutoEncoder class to compress the input data and encoded input neurons to 25. Next, we can use compressed data as new input data to train the neural network model, and test new compressed data. This method can improve the model accuracy, and detailed results will be discussed in the part 3.

2.5 Genetic Algorithm for Feature Selection

High dimensional data often contains useless and noise features. Feature selection is to select a subset features which are more essential and can be used to decrease search space size and improve performance and effectiveness [11]. Feature selection can act as an important role to remove noise features. By using smaller subset features, it also can preserve most of original data information and decrease cost time [12].

Genetic algorithm is inspired by evolution and natural selection, which can be used for feature selection [13]. First, we need to initialize population size, DNA size and generate the population. For our dataset, the input neurons are 32, so the DNA size is 32. We experiment different population size from 5 to 100 to get different results and compare to select better useful features. Every population index value is binary. If this feature is selected, the index value is 1, otherwise the value is 0. Next step is to define the fitness. We use the accuracy to evaluate the fitness value. The individual with higher accuracy will be selected. If those features combination can get higher accuracy, we can think that they are more essential features. After selecting better individuals, we can view them as parents and process the crossover step to produce their children. For every child, some features from one parent and rest of feature from another parent. We set cross rate is 0.8 in our model. The crossover step may generate children that are very similar to their parents, to solve this problem, and we can use the mutation method [14]. For every individual, if every point's random value is less than mutation probability, which we set to 0.002, this point will be mutated. If the point value is 1, the mutated value is 0. These steps will iterate many generations to select better features.

3 Results and Discussion

After implementing basic neural network, we apply the autoencoder method and compare their results. The detailed result is shown in Table 2. The basic three layers' neural network model can get 92% accuracy and 0.9 f score. After applying autoencoder method, it can get the highest 96% accuracy and 0.95 f score. The reason is that after using autoencoder method, it can get more compressed input data, remove some duplicated or useless features. Our dataset is very small about 400 instances, but there are 32 attributes. It's very likely to have many noisy features. Thus, after using autoencoder technology, we can get better result than our basic neural network.

Table 2. Basic Neural Network (NN) and AutoEncoder Results

	Basic NN	Basic NN + AutoEncoder
Test Accuracy	92%	90% ~ 96%
F-score	0.90	0.87 ~ 0.95

For the genetic algorithm, we have experimented different population size, including 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. For every population size, we can get one most fitted DNA. Every DNA has 32 attributes, which from 0 to 31. Besides, we will iterate many iterations. If our iteration epoch is 5, we can get 5 most fitted DNA for every population size. For every population size, we can count the number of every feature in the 5 most fitted DNA. If the feature count is larger than 3, we can choose these features. Then, we can get a selected feature list for every population size. There are 11 different population size, including 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100. Next, we also count the number of each feature in these selected 11 feature lists to choose better features. The count of each feature is shown in Figure 1. Next, we select features by using their count number, such as larger than 5, 6, 7, 8, 9, 10. Then, we use selected features to train neural network model and test the testing dataset. The detailed compared accuracy is shown in Table 3. From this table, if count value is equal and larger than 6, with the increase of the count, the test accuracy is decreasing. It's important to remove appropriate number of features. The best test accuracy is 95% and f score is 0.93 when feature count is larger than 5 (count >= 6). We remove redundant seven features, including the attribute 4, 10, 11, 19, 20, 25, 28, and keep other 25 features. Compared with our basic neural network model, the model applies feature selection by using genetic algorithm can get better result. After selecting features, we can remove those noisy features, get better compressed representation of data, decrease the cost time, and also preserve most original dataset. The original 32 features have some duplicated features and result in little overfitting. Also, we can't remove too many features, which may lead to lose some important information. Selecting appropriate number of features is very important.

From Table 4 result, the neural network after applying genetic algorithm can get better test accuracy and f score than the basic neural network. After feature selection, the model can get more useful and important features to train the model. Some noise and redundant features influences the basic neural network model performance.

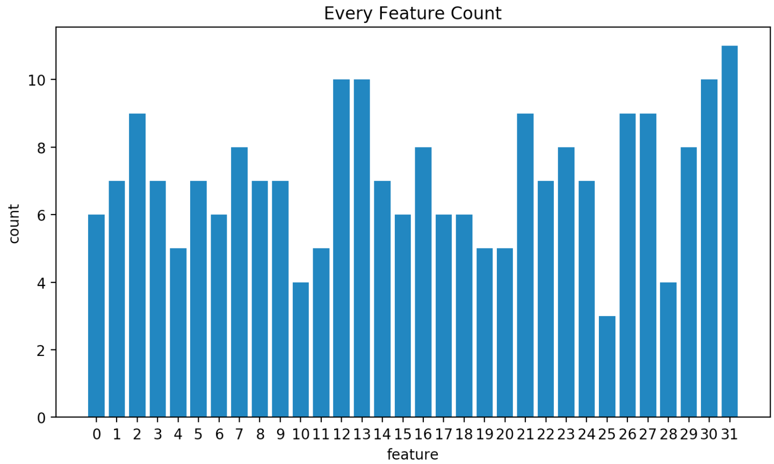

Fig. 1. The x axis is feature, and the y axis the feature count.

Table 3. Compare Different Feature Count Result

	Count >= 5	Count >= 6	Count >= 7	Count >= 8	Count >= 9	Count >= 10
Training Accuracy	96%	96%	95%	94%	91%	91%
Test Accuracy	94%	95%	94%	94%	94%	86%
F-score	0.92	0.93	0.92	0.92	0.92	0.78

Table 4. Basic Neural Network (NN) and Genetic Algorithm (GA) Results

	Basic NN	Basic NN + GA
Test Accuracy	92%	95%
F-score	0.90	0.93

For AutoEncoder method and genetic algorithm, they can both improve the neural network. They also compress the input data and remove some noisy features. We can use these two methods to process our data. Comparing these two methods, after applying AutoEncoder technology, the accuracy can achieve accuracy between 90% and 96%. Although the result is not very stable, it runs very fast and achieves good result. In our computer, it only runs several seconds. However, for the genetic algorithm, if we set generation size and population to large number, it will run several minutes. The genetic algorithm will take longer time, but this technology is very useful, can improve the accuracy and get a stable accuracy. Overall, these two methods can improve the test accuracy and f score, and the detailed result shown in Table 5.

Table 5. Basic Neural Network (NN), AutoEncoder and Genetic Algorithm (GA) Results

	Basic NN	Basic NN + AutoEncoder	Basic NN + GA
Test Accuracy	92%	90% ~ 96%	95%
F-score	0.90	0.87 ~ 0.95	0.93

Many papers have been used Neural Network model. Paulo and Alice used the same students' performance dataset and neural network model, and they got the accuracy between 87.6% and 89% [3]. Our model gets better result than theirs, and the basic model can achieve 92% accuracy. The detailed result is shown in Table 6. The compared paper didn't give detailed parameters value. The reason why our basic NN model is better is that we have experienced many hyper parameters and find better fitted value, such as different learning rate, number of hidden layer neurons, number of epochs, different activate function, different loss function and optimizer. For example, for the learning rate, we have tried 0.001, 0.01, 0.1, etc. Besides, we may have different number of training and test dataset. The Paulo's paper didn't mention how much data is using for training and testing. Overall, our testing accuracy is very similar, and is not very big difference. For the evaluation method, Paulo and Alice only used accuracy for classification problem [3]. However, we think for some unbalanced data, accuracy is not enough, so we also use f-score to evaluate my model. For the basic neural network model, the f-score is 0.90.

Table 6. Paulo's NN and Our Basic NN Results

	Paulo's NN	Our Basic NN
Accuracy	87.6% ~ 89%	92%
F-score	/	0.90

4 Conclusion and Future work

Predicting students' performance is very useful for schools, educators and students. This paper uses the read-world student dataset and builds a model to classify whether the student can pass or fail the final exam automatically. We implement the basic neural network by using PyTorch, which has three layers and can achieve 92% accuracy and 0.90 f score. Comparing with other published paper, the model can get better accuracy score. To improve the basic neural network model, we apply autoencoder method to get more compressed dataset, and can achieve the highest 96% accuracy and 0.95 f score. To remove redundant and noisy feature, this paper applies the genetic algorithm for feature selection, which can improve the accuracy to 95% and f score to 0.93 compared to the basic neural network model. Those two methods are good ways to improve the neural network model.

Besides, there is some work to be extended. First, sometimes a single classifier can't get a good result in all situations, and different classifier, such as support vector machine, k-nearest neighbor, naïve bayes and decision tree, works well in different case. Thus, we can try different models and combine several classifiers to improve the prediction accuracy. Secondly, we can get more data and try deep learning method to improve the model performance. Finally, now the model uses the off-line data, more and more students like to attend on-line courses, so we can train the model to predict the online student performance in real time.

5 References

[1] A. Shahiri, W. Husain and N. Rashid, "A Review on Predicting Student's Performance Using Data Mining Techniques", Procedia Computer Science, vol. 72, pp. 414-422, 2015.

[2] S. Kotsiantis, K. Patriarcheas and M. Xenos, "A combinational incremental ensemble of classifiers as a technique for predicting students' performance in distance education", Knowledge-Based Systems, vol. 23, no. 6, pp. 529-535, 2010.

[3] P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008.

[4] J. Sujatha and S. P. Rajagopalan, "Performance Evaluation of Machine Learning Algorithms in the Classification of Parkinson Disease Using Voice Attributes", International Journal of Applied Engineering Research, vol. 12, no. 21, pp. 10669-10675, 2017.

[5] K. J. Danjuma, "Performance evaluation of machine learning algorithms in post-operative life expectancy in the lung cancer patients", 2015.

[6] B. Sturm, "Classification accuracy is not enough", Journal of Intelligent Information Systems, vol. 41, no. 3, pp. 371-406, 2013.

[7] D. M. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation", 2011.

[8] T. Gedeon and D. Harris, "Progressive Image Compression", Proceedings International Joint Conference on Neural Networks, vol. 4, pp. 403-407, 1992.

[9] P. Surmenok, "Estimating an Optimal Learning Rate For a Deep Neural Network", Towards Data Science, 2017. [Online]. Available: https://towardsdatascience.com/estimating-optimal-learning-rate-for-a-deep-neural-network-ce32f2556ce0.

[10] S. Wang, Z. Ding and Y. Fu, "Feature Selection Guided Auto-Encoder", AAAI, pp. 2725-2731, 2017.

[11] K. Han, Y. Wang, C. Zhang, C. Li and C. Xu, "Autoencoder Inspired Unsupervised Feature Selection", 2017.

[12] H. Vafaie and K. Jong, "Genetic Algorithms as a Tool for Feature Selection in Machine Learning", Int. Conf. on Tools with AI, pp. 200-203, 1992.

[13] A. Aziz, A. Azar, M. Salama, A. Hassanien and S. Hanafy, "Genetic Algorithm with Different Feature Selection Techniques for Anomaly Detectors Generation", in Federated Conference, 2013, pp. 769-774.

[14] F. Gómez and A. Quesada, "Genetic algorithms for feature selection in Data Analytics | Neural Designer", Neuraldesigner.com, 2018. [Online]. Available: https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection.

6 Appendix

Table 7. Dataset attributes information [3]

Attribute	Description	Domain	Value	Class
school	student's school	binary: "GP" - Gabriel Pereira or "MS" - Mousinho da Silveira	"GP" "MS"	0 1
sex	student's sex	binary: "F" - female or "M" - male	"F" "M"	0 1
age	student's age	numeric: from 15 to 22	string: from 15 to 22	numeric: from 15 to 22
address	student's home address type	binary: "U" - urban or "R" - rural	"U" "R"	0 1
famsize	family size	binary: "LE3" - less or equal to 3 or "GT3" - greater than 3	"LE3" "GT3"	0 1
Pstatus	parent's cohabitation status	binary: "T" - living together or "A" - apart	"T" "A"	0 1
Medu	mother's education	numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education	string: from 0 to 4	numeric: from 0 to 4
Fedu	father's education	numeric: 0 - none, 1 - primary education (4th grade), 2 – 5th to 9th grade, 3 – secondary education or 4 – higher education	string: from 0 to 4	numeric: from 0 to 4
Mjob	mother's job	nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other"	"teacher" "health" "services" "at_home" "other"	0 1 2 3 4
Fjob	father's job	nominal: "teacher", "health" care related, civil "services" (e.g. administrative or police), "at_home" or "other"	"teacher" "health" "services" "at_home" "other"	0 1 2 3 4
reason	reason to choose this school	nominal: close to "home", school "reputation", "course" preference or "other"	"home" "reputation" "course" "other"	0 1 2 3
guardian	student's guardian	nominal: "mother", "father" or "other"	"mother" "father" "other"	0 1 2
traveltime	home to school travel time	numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour	string: from 1 to 4	numeric: from 1 to 4
studytime	weekly study time	numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours	string: from 1 to 4	numeric: from 1 to 4
failures	number of past class failures	numeric: n if 1<=n<3, else 4	string: from 1 to 4	numeric: from 1 to 4
schoolsup	extra educational support	binary: yes or no	"yes" "no"	0 1
famsup	family educational support	binary: yes or no	"yes" "no"	0 1
paid	extra paid classes within the course subject (Math or Portuguese)	binary: yes or no	"yes" "no"	0 1
activities	extra-curricular activities	binary: yes or no	"yes" "no"	0 1
nursery	attended nursery school	binary: yes or no	"yes" "no"	0 1
higher	wants to take higher education	binary: yes or no	"yes" "no"	0 1
internet	Internet access at home	binary: yes or no	"yes" "no"	0 1
romantic	with a romantic relationship	binary: yes or no	"yes" "no"	0 1
famrel	quality of family	numeric: from 1 - very bad to 5 - excellent	string: from 1 to 5	numeric: from 1 to 5

	relationships			
freetime	free time after school	numeric: from 1 - very low to 5 - very high	string: from 1 to 5	numeric: from 1 to 5
goout	going out with friends	numeric: from 1 - very low to 5 - very high	string: from 1 to 5	numeric: from 1 to 5
Dalc	workday alcohol consumption	numeric: from 1 - very low to 5 - very high	string: from 1 to 5	numeric: from 1 to 5
Walc	weekend alcohol consumption	numeric: from 1 - very low to 5 - very high	string: from 1 to 5	numeric: from 1 to 5
health	current health status	numeric: from 1 - very bad to 5 - very good	string: from 1 to 5	numeric: from 1 to 5
absences	number of school absences	numeric: from 0 to 93	string: from 0 to 93	numeric: from 0 to 93
G1	first period grade	numeric: from 0 to 20	string: from 0 to 20	numeric: from 0 to 20
G2	second period grade	numeric: from 0 to 20	string: from 0 to 20	numeric: from 0 to 20
G3	final grade	numeric: from 0 to 20, output target	string: from 0 to 20	1 (if numeric value >= 10) 0 (if numeric value < 10)