Stacking Multiple Machine Learning Ways and Using Optuna to Precisely Predict Students' GPA

Chuyi Qu

International School of Information and Software, Dalian University of Technology, Dalian, China

quchuyi123@mail.dlut.edu.cn

Abstract. The application of the Internet in the field of education and teaching is increasingly widespread, and there are massive educational data generated in this process. How to make reasonable use of these massive educational data has always been an important issue in the field of educational data mining. A student's Grade Point Average (GPA) is crucial for evaluating their own development, helping teachers plan teaching, and enabling schools to formulate education programs. Although there have been many precedents of using machine learning to predict students' GPA, the fitting process is often relatively simple. The method adopted in this study, which comprehensively utilizes Stacking and Optuna. Tune the hyperparameters of the base models using Optuna to enhance their fitting capabilities, successfully leverages the advantages of both, the coefficient of determination (R^2) reaches 0.88. And this way is more accurate than previous model construction methods, demonstrating the potential and prospects of this method in the field of regression fitting.

Keywords: GPA; Stacking; Optuna: Catboost; LightGBM.

1. Introduction

GPA is an important criterion for measuring students' academic performance. Most schools use it as the primary factor for evaluating scholarships and selecting outstanding students. Meanwhile, GPA is also an indispensable part of the resume for further education. Therefore, studying students' behaviors, such as study duration and online entertainment time, to accurately predict their GPA can alert students to identify their shortcomings or find ways to optimize themselves. For schools, it can help better design courses and revise teaching plans, thereby promoting the improvement of students' GPA.

To predict the final GPA based on students' behaviors, this paper adopts machine learning methods. Machine learning has significant advantages in processing large datasets and performing model regression and classification. It enables machines to continuously learn from a large number of test sets and use the final model to predict students' GPA. At present, there have been many related studies. For example, the article uses CatBoost, a gradient boosting tree, to predict students' academic performance [1]. CatBoost, as an algorithm, natively supports categorical features, which means that for categorical features in the dataset, there is no need for preprocessing methods such as One-Hot Encoding or Ordinal Encoding [2]. There are also many categorical features in students' data, such as 'gender' and 'major', so catboost is suitable. Another example is that in paper [3], Lakkaraju et al. trained models using methods such as Support Vector Machine (SVM), Random Forest, Logistic Regression, Adaboost, and Decision Tree. Additionally. Bayesian optimization is used to tune XGBoost [4]. What is noteworthy is that the tuning method adopted in this study is Optuna, which is an automated integrated tuning tool that includes various tuning methods such as Bayesian optimization. This paper aims to find more complex and accurate methods. Firstly, it uses Optuna to adjust the hyperparameters in CatBoost to improve model accuracy and enhance fitting [5]. Inspired by the paper [6], it also uses the stacking method to strive for a stronger fitting model under the combination of different base models that have been tuned by Optuna. At the same time, it is necessary to control the problem of overfitting to ensure that the model has stronger applicability.

2. Research Methods

2.1. CatBoost

This study first adopts the CatBoost model. As a type of gradient boosting tree, this model has a notable advantage over other gradient boosting tree models: it does not require data preprocessing methods such as One-Hot Encoding for categorical features. Since there are many categorical features involved in this dataset, this method is prioritized. The specific principles and steps of this method are as follows:

Initialize a weak learner, usually a decision tree, denoted as $f_0(x)$, whose prediction result is the initial predicted value $\hat{y}_0$. At this point, there is a deviation between the initial predicted value and the true value.

Calculate the residual, which is the difference between the true value y_i and the current model's predicted value $\hat{y}_i$.

$$r_i = y_i - \hat{y}_i \tag{1}$$

Construct a decision tree: use the calculated residuals as the new target values, and build a new decision tree $f_t(x)$ in a "symmetric tree structure" manner. Meanwhile, adopt some regularization techniques such as limiting the depth of the decision tree and controlling the number of leaf nodes.

Update the model based on the newly trained decision tree:

$$y_{i,t} = \hat{y}_{i,t-1} + \alpha f_t(x_i) \tag{2}$$

Repeat the above steps, continuously training new decision trees and updating the model until the preset or default conditions are met. Finally, the CatBoost model consists of multiple decision trees, and its prediction result is the combined effect of all these decision trees.

2.2. LightGBM

The researchers then attempted the LightGBM model [7]. Also as a gradient boosting tree model, it adopts a histogram algorithm and an innovative leaf-wise growth strategy.

Histogram Algorithm: Traditional GBDT needs to sort all sample values of each feature and traverse them to find the optimal split point when selecting split points. LightGBM discretizes continuous features into a limited number of "bins" through "histograms", which significantly reduces the cost of searching for split points.

The algorithm steps are as follows:

Initialize the model, taking the mean value of all samples as the initial predicted value.

Calculate the negative gradient, which represents the "error of the current model" and serves as the fitting target for the new tree.

Construct the tree structure using histogram binning: divide the feature values into k bins (e.g., $[x_1, x_2), [x_2, x_3), \ldots$), and count the number of samples and the sum of residuals in each bin:

$$\text{count}_b = \sum_{i \in \text{bin } b} 1, \quad \text{sum}_b = \sum_{i \in \text{bin } b} r_i^{(m)} \tag{3}$$

Calculate the split gain and select the split point with the maximum gain, then recursively split the leaf nodes.

Update the model prediction: The predicted value of the m-th tree is the weight $w_j^{(m)}$ of the leaf node (calculated by minimizing the loss). The final predicted value is accumulated as $\hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + \eta \cdot w_{j(i)}^{(m)}$ (where η is the learning rate, controlling the contribution of each tree; $j(i)$ is the index of the leaf where sample i falls in the m-th tree)

Stop iteration when the maximum number of trees is reached or the loss no longer decreases, and output the final model:

$$\hat{y}_i = \hat{y}_i^{(0)} + \eta \sum_{m=1}^{M} w_{j(i)}^{(m)} \tag{4}$$

2.3. Optuna

Based on Bayesian Optimization, Optuna constructs a probabilistic model to predict the performance of hyperparameter combinations, thereby efficiently exploring the hyperparameter space. Compared with traditional Grid Search or Random Search, Optuna is more intelligent and can find better hyperparameter combinations with fewer trials.

2.4. Stacking

Stacking is an ensemble learning method that combines the prediction results of multiple base learners to train a higher-level model (meta-learner), thereby achieving better prediction performance than a single model. Its core idea is to leverage the complementarity of different algorithms in processing data and explore the features and patterns of data through multi-layer learning to achieve more accurate predictions. The steps are as follows:

Train multiple base learners: Select several different learning models to train multiple base learners, denoted as

$$h_1, h_2, ..., h_K \tag{5}$$

Construct a new training set: Take the predicted values of each base learner on the training set as a new feature matrix, and combine it with the labels of the original training set to form a new training set.

Train the meta-learner: Select a learning algorithm to train a learner on the new training set, i.e., the meta-learner. The expression is:

$$H(X) = F\big(h_1(X), h_2(X), ..., h_K(X)\big) \tag{6}$$

3. Result

3.1. Dataset

The dataset used in this study is an open-source one from Kaggle, which simulates the academic and lifestyle behaviors of 80,000 students, including various features such as study habits, mental health, family background, motivation, and environmental factors. It is a synthetic dataset generated using Python, with realistic statistical modeling, Gaussian distribution, conditional logic, and heuristic methods to simulate actual student behaviors and academic performance. The dataset contains 30 independent variables, among which 12 are categorical features and the remaining 18 are numerical features.

3.2. Evaluation Metrics

This study uses Root Mean Square Error (RMSE) and Coefficient of Determination (R^2) as evaluation metrics.

RMSE is "the square root of the average of the squares of the errors". It amplifies the impact of larger errors through squaring, and then restores to the same order of magnitude as the original data through square root, thus being more sensitive to outliers with larger errors [8]. The lower its value, the better. The formula for RMSE is:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{7}$$

R^2 is a core indicator for measuring the fitting effect of regression models, with a value range between 0 and 1 [9]. The higher the value, the stronger the fitting ability of the model. Its calculation formula is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (8)$$

3.3. Analysis of Experimental Results

First, this experiment trained using the original CatBoost and LightGBM. As shown in table 1, opt represents optuna, c represents catboost, l represents lightgbm, rf represents random forest. With CatBoost in its default (non-hyperparameter-tuned) state, the model achieved an RMSE of 4.08 and an R^2 of 0.8796, demonstrating excellent fitting ability. When using the LightGBM model, the RMSE was 4.09 and the R^2 value was 0.8793, which was slightly inferior to CatBoost but still showed good fitting performance. Additionally, its training speed was significantly faster than that of CatBoost.

To further improve the model's fitting ability, Optuna was used to optimize the hyperparameters of LightGBM and CatBoost, aiming to achieve a higher R^2 value.

The results of tuning CatBoost with Optuna were: an RMSE of 4.08 and an R^2 of 0.8799.

The results of tuning LightGBM with Optuna were: an RMSE of 4.08 and an R^2 of 0.8797.

It can be seen that after tuning with Optuna, the accuracy of the models was further improved.

Subsequently, the stacking method was also attempted [10]. When using CatBoost, LightGBM, Random Forest, and SVR as base learners, and Linear Regression as the meta-learner, the RMSE was 4.08 and the R^2 was 0.8797. Thus, the initial attempt at stacking did not yield better results than the optimized CatBoost, and its speed was also slower than that of CatBoost tuned with Optuna [11]. However, with the base learners and meta-learner remaining unchanged, when Optuna was used to tune the hyperparameters of CatBoost alone, the results were: an RMSE of 4.08 and an R^2 of 0.87996. When, with the same base learners and meta-learner, Optuna was used to tune the hyperparameters of CatBoost, LightGBM, and Random Forest simultaneously, the results were: an RMSE of 4.08 and an R^2 of 0.8800.

Table 1: Performance comparison of different models

	Catboost	LightGBM	Catboost(opt)	LightGBM(opt)	Stacking	Stacking (opt[c])	Stacking (opt[c/l/rf])
RMSE	4.08	4.09	4.08	4.08	4.08	4.08	4.08
R^2	0.8796	0.8793	0.8799	0.8797	0.8797	0.87996	0.8800

4. Challenges and Prospects

Perhaps limited by the dataset itself, it is difficult to achieve a coefficient of determination (R^2) above 0.88. When this paper further optimized the meta-learner using Optuna, it instead led to overfitting, resulting in a lower R^2. Therefore, it is hoped that in the future, methods can be found to overcome overfitting while maximizing the model's accuracy. This paper only explored methods to fit the dataset without conducting research on the conclusions. In the future, visualization methods can be used to find an intuitive connection between students' behaviors and GPA.

5. Conclusion

The new model training method summarized in this study, which comprehensively utilizes Optuna and stacking, enables the model to achieve an R^2 of 0.88, effectively improving the model's fitting ability. Researchers can use the model trained by this method to predict students' GPA, or conduct visualization processing to explore the relationship between influencing factors and results, the importance of factors, etc., providing accurate guarantees for solving practical problems. Moreover, researchers believe that this method is not only useful for this dataset but will also have

advantages when applied to other datasets, and will be one of the effective ways to improve model accuracy.

References

[1] A. Joshi, P. Saggar, R. J. Moolchand, S. Deepak, G. A. Khanna. “CatBoost - An Ensemble Machine Learning Model for Prediction and Classification of Student Academic Performance.” Advances in Data Science and Adaptive Analysis: Theory and Applications, 13(4):2141002-1-2141002-28 (2021).

[2] L. Prokhorenkova, G. Gusev, A. Vorobev, et al. “CatBoost: unbiased boosting with categorical features.” Advances in neural information processing systems, 31. (2018).

[3] B. Albreiki, N. Zaki, H. Alashwal. “A Systematic Literature Review of Student’ Performance Prediction Using Machine Learning Techniques.” Educ. Sci. 11, 552. (2021).

[4] R. Arifuddin, et al. "Effectiveness of Machine Learning Models with Bayesian Optimization-Based Method to Identify Important Variables that Affect GPA." Jurnal Teknologi dan Aplikasi Matematika (JTAM), vol. 8, no. 3, art. 21711. (2023).

[5] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. “Optuna: A Next-generation Hyperparameter Optimization Framework.” In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). Association for Computing Machinery, New York, NY, USA, 2623–2631. (2019).

[6] J. M. Ahn, J. Kim, K. Kim. “Ensemble Machine Learning of Gradient Boosting (XGBoost, LightGBM, CatBoost) and Attention-Based CNN-LSTM for Harmful Algal Blooms Forecasting.” Toxins. 15, 608. (2023).

[7] G. Ke, Q. Meng, T. Finley, et al. “Lightgbm: A highly efficient gradient boosting decision tree.” Advances in neural information processing systems, 30. (2017).

[8] T. O. Hodson. “Root means square error (RMSE) or mean absolute error (MAE): When to use them or not.” Geoscientific Model Development Discussions, 1-10. (2022).

[9] N. J. D. Nagelkerke. “A note on a general definition of the coefficient of determination.” biometrika, 78(3): 691-692. (1991).

[10] J. R. Yu, X. M. Chang, S. H. Hu, H. D. Yin, J. J. Wu. “Combining travel behavior in metro passenger flow prediction: A smart explainable Stacking-Catboost algorithm,” Information Processing & Management, Volume 61, Issue 4, 103733, ISSN 0306-4573, (2024).

[11] S. Džeroski, B. Ženko. “Is combining classifiers with stacking better than selecting the best one?” Machine learning, 54(3): 255-273. (2004).