Practical Strategies for Improving the Performance of Student Response Prediction

Daichi Takehara
Aidemy Inc.
Tokyo, Japan
takehara-d@aidemy.co.jp

Yuto Shinahara
Aidemy Inc.
Tokyo, Japan
shinahara-y@aidemy.co.jp

Abstract

This report presents a solution to the NeurIPS 2020 education challenge, which is positioned in first place in task 1 and second place in task 2. Our solution is based on useful feature extraction through target encoding, time-related features, user history features, subject features, and several strategies to achieve high accuracy, including feature selection and meta features. The code for the solutions is available at https://github.com/haradai1262/NeurIPS-Education-Challenge-2020.

1 Introduction

With the widespread use of online education systems, educational data mining technologies, which extract useful information from educational data and utilize it to improve student learning outcomes, are becoming increasingly important. A diagnostic question, representing a multiple choice question with four possible answers, has attracted much attention in this research area. The selection of incorrect answers by students learning the diagnostic questions reveals something about their misconceptions and is a key for understanding students' knowledge.

The NeurIPS 2020 education challenge [1] was held. The competition focuses on students' records of their answers to multiple-choice diagnostic questions in an educational system. It sets tasks to predict whether students answer correctly (task 1) and which choices they answer (task 2). These tasks can impact education through applications such as recommending questions at an appropriate level of difficulty that best fit the student's background and uncovering potential common misconceptions that students may have.

This report presents a solution for task 1 and task 2 of the NeurIPS 2020 education challenge. The main contributions to the performance of the solutions are as follows:

- We extracted useful features for student response prediction. The features derived from the students' age, learning experience, and the subjects worked well. To extract these features, target encoding [2, 3] was useful.
- We introduce several strategies to improve accuracy. We removed redundant features through feature selection based on feature importance. Furthermore, by inputting meta features extracted from the prediction models' output into other prediction models, we solved task 1 and task 2 cooperatively.

2 Challenge

2.1 Dataset

Task 1 and task 2 provides a dataset of 15,867,850 answers by 118,971 students to 27,613 multiple-choice diagnostic questions. The dataset is provided by the online education provider Eedi, which

1st NeurIPS Education Challenge (NeurIPS 2020), Vancouver, Canada.

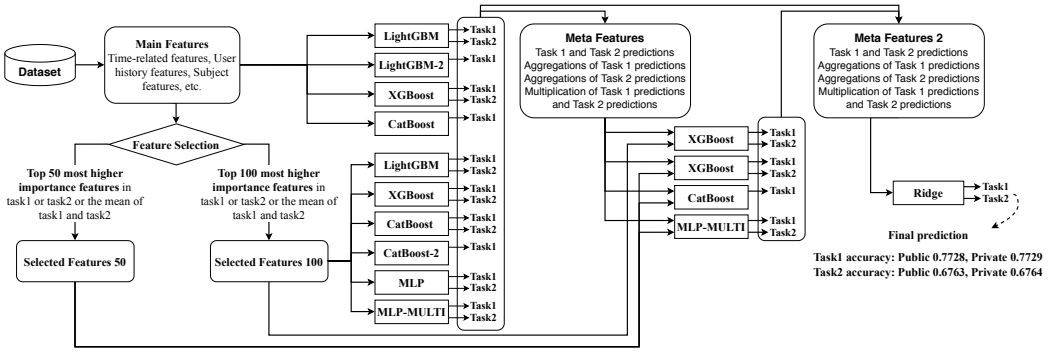

Figure 1: Overview of our solution. First, we extracted features from the dataset. The obtained features were used to train the models for task 1 and task 2. Next, redundant features were removed through feature selection, and new models were trained using the selected features. Furthermore, meta features were extracted from the models' predictions and inputted into new models. An ensemble of several models obtained the final prediction results.

offers crowdsourced diagnostic questions for elementary school students through high school, between September 2018 and May 2020[1]. More details of the dataset can be found in [1].

2.2 Task Description

Task 1 is to predict whether a student answers a question correctly,—a binary classification problem. In task 1, prediction accuracy (#correct predictions / #total predictions) is used as an evaluation metric. To predict whether a student answers unknown questions correctly is crucial for estimating the student's ability level in education systems and forms the groundwork for more advanced tasks.

Task 2 is to predict which choices a student answers,—a multi-class classification problem. Task 2 also uses prediction accuracy, which is the same metric as above except that the true answers are categorical instead of binary. Predicting the actual multiple-choice option for a student's answer allows the analysis of likely common misconceptions that a student may hold on a topic and thus forms the basis for personalized advice and guidance on education systems.

3 Method

3.1 Features

An overview of the solution is shown in Figure 1. First, we extract useful features that effectively represent students and problems from the dataset. This report describes only those features that contribute significantly to the performance among the extracted features. Please refer to the published code for strict processing or other features.

Target Encoding First, we describe target encoding [2, 3], a technique we have utilized for many feature extractions in our solution. Target encoding calculates statistics from a target variable segmented by the unique values of one or more categorical features. The target variables are labels indicating whether the user answered correctly (IsCorrect) for task 1 and the user's answered value (AnsweredValue) for task 2. Target encoding is well known as a useful feature extraction technique in machine learning competitions. In both task 1 and task 2 that we worked on, it was expected that the features obtained by target coding would also be intuitively important, such as the user's average percentage of correct answers and which options were more likely to be selected for a question.

Specifically, we smoothed the target encoding features to avoid overfitting for low-frequency categories as in [3] as follows:

$$TE_{target}([Categories]) = \frac{count([Categories]) * mean_{target}([Categories]) + w_{smoothing} * mean_{target}}{count([Categories]) + w_{smoothing}} \quad (1)$$

[1] `https://eedi.com/`

where $count(\cdot)$ is the number of observations for that category, and $w_{smoothing}$ is a smoothing hyperparameter, which we set to 5. $[Categories]$ is a list of categorical features whose values are tuples with combinations of features that occur in the dataset.

Time-Related Features The dataset includes timestamp of answered time and users' date of birth. We extracted the day, hour, weekday, and week of the month of a students' answered time. Students' ages were also extracted. Student age features effectively worked in combination with other attributes. For example, questions' correctness with each age could be expressed by combining them with question id and applying target encoding.

User History Features We extracted features about the users' learning history. More specifically, we extracted features that represent students' experience to date, such as the number of students' answers, the number of quizzes, the number of classes to which questions were assigned, the number of subject types answered, and the duration of the course. We also extracted features representing the most recent situation, such as the most recent answer's elapsed time.

Subject Features The subjects are in the area of mathematics, including topics such as "Algebra," "Data and Statistics," and "Geometry and Measure." These are essential concepts for students' knowledge states. One of the features that worked well in our solution was features obtained by decomposing into the student-subject matrix, which is derived from the students' answer records, using singular value decomposition [4]. The features can represent the relationship between the subjects by the co-occurrence of the students' answers. Other features that worked well were max, min, and standard deviation aggregations of target encoding values by the combination of each user id and subject id. The features allow us to represent the degree to which students are good or poor at a particular subject included in a question.

3.2 Modeling

Next, we describe several strategies we have implemented to improve the performance of our prediction models.

First, we created a set of five fold-out validation sets. We conducted a stratified sampling of each fold-out in the validation sets to get the same ratio for user id and IsCorrect, known as stratified k-fold cross-validation.

We used a gradient boosting decision tree (GBDT) and multilayer perceptron (MLP) as algorithms to train the prediction models. Here, we predict the task 1 target and task 2 target in the same manner. Using the same features and algorithms, when solving Task1, IsCorrect is set as the target of binary classification. When solving task 2, AnswerValues is set as the target of multi-class classification, and only the target is changed to create a prediction model. We used well-known implementations of GBDT, LightGBM [5], XGBoost [6], and CatBoost [7]. We expected that the use of multiple models would result in robust results when ensembled. We used an MLP that includes batch normalization [8] and dropout [9]. Unlike GBDT, the numerical features were applied to min-max scaling and converted to a scale from zero to one. Also, we transformed categorical features into low-dimensional vectors using the embedding layers. We also used a multi-task MLP (MLP-MULTI) that optimized loss function for both task 1 and task 2 in a single training session.

Feature Selection To reduce redundant features and train the models effectively, we introduced feature selection. We used the feature importance implemented in LightGBM. The importance-type parameter was set to "gain,' which can obtain the total gains of splits that use the feature. In the solution, the importance of the task 1 model, task 2 model, and their mean values were used to obtain the top 100 features (selected features 100) and the top 50 features (selected features 50), respectively. A subset of the features obtained by feature selection is used to train a new prediction model.

Meta features We use the models' predicted values to extract the new features (meta features) to input other models. The predicted value of task 1 is the probability value of predicting IsCorrect; the predicted value of task2 is the 4-dimensional probability value of AnsweredValue. In addition to the predictions themselves, max, min, mean, standard deviation aggregation of multiple models' predicted values are also used. The mean predicted values of task1 models multiplied by the mean predicted values of task2 models are also used. It is expected that the meta features can effectively

Table 1: Results of the prediction models included in our solution.

Features	Model	Task 1 Accuracy	Task 2 Accuracy
Main Features	LightGBM	0.76767	0.67026
Main Features	LightGBM-2	0.76760	—
Main Features	XGBoost	0.76778	0.67075
Main Features	CatBoost	0.76807	—
Selected Features 100	LightGBM	0.76793	0.67081
Selected Features 100	XGBoost	0.76844	0.67107
Selected Features 100	CatBoost	0.76806	0.66862
Selected Features 100	CatBoost-2	0.76867	—
Selected Features 100	MLP	0.76272	0.66659
Selected Features 100	MLP-MULTI	0.76280	0.66653
Selected Features 50 + Meta Features	XGBoost	0.76952	0.67239
Selected Features 100 + Meta Features	XGBoost	0.76954	0.67248
Selected Features 50 + Meta Features	CatBoost	0.76954	—
Selected Features 50 + Meta Features	MLP-MULTI	0.77014	0.67281
Meta Features 2	Ridge	0.77023	0.67293

transfer the potential representation of the students' ability obtained from task 1 to task 2 and the state of the students' knowledge obtained from task 2 to task 1.

4 Results

Table 1 shows the results of the models included in our solution. These models combine several features and strategies described in the previous sections. From these results, we can see the gains to the accuracy of each strategy.

First, the accuracy of the prediction model using GBDT with all the features is already 0.76807 for task 1 and 0.67075 for task 2 (the difference from the final accuracy is about 0.002 in both tasks). Therefore, it can be confirmed that feature extraction in the solution makes an outstanding contribution. It should be noted that many of the features that occupy the top of the prediction model's feature importance are those to which the target encoding is applied.

Next, we confirmed the model using selected features 100 that performed feature selection. With the model using the main features, the accuracy improvement of task 1 is 0.0008 and task 2 is 0.0003, and it was confirmed that removing redundant features contributes to accuracy improvement. This process was also valuable from the viewpoint of efficiency, shortening the learning time.

Furthermore, the accuracy of the model using meta features can be confirmed to be improved by 0.0017 for task 1 and 0.0018 for task 2 from the model using selected features 100. We believe that this accuracy improvement is due to the effective transfer of the potential features in task 1 to task 2 and the potential features in task 2 to task 1.

Finally, the final prediction results were obtained from the ensembles of many models. Stacking ridge regression [10], a method of blending each model's prediction results by a linear sum based on the weights learned by ridge regression, was adopted as the ensemble method. As a result, the score of task 1 was public 0.7728, private 0.7729, and the score of task 2 was public 0.6763, private 0.6764 on the leaderboard, and they could be positioned as first place and second place, respectively.

5 Conclusion

In this report, we presented a solution for task 1 and task 2 of the NeurIPS 2020 education challenge. Our solution was based on useful feature extraction, that is, target encoding, time-related features, user history features, subject features, and several strategies to achieve high accuracy, —feature selection, meta features. As a result, we were able to archive task 1 in first place and task 2 in second place in the challenge.

References

[1] Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, José Miguel Hernández-Lobato, Richard E Turner, Richard G Baraniuk, Craig Barton, Simon Peyton Jones, et al. Diagnostic questions:the neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061*, 2020.

[2] Daniele Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1):27–32, 2001.

[3] Benedikt Schifferer, Gilberto Titericz, Chris Deotte, Christof Henkel, Kazuki Onodera, Jiwei Liu, Bojan Tunguz, Even Oldridge, Gabriel De Souza Pereira Moreira, and Ahmet Erdem. Gpu accelerated feature engineering and training for recommender systems. In *Proceedings of the Recommender Systems Challenge 2020*, pages 16–23. 2020.

[4] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[5] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

[6] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD Int. Conf.on Knowledge Discovery & Data Mining*, pages 785–794, 2016.

[7] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, pages 6638–6648, 2018.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd Int. Conf. on Machine Learning*, pages 448–456, 2015.

[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[10] Leo Breiman. Stacked regressions. *Machine learning*, 24(1):49–64, 1996.