STATS 503 Kaggle Report

Team Name: Tongyao Team
Yuxiang Gao, Chenfei Wang, Tongyao Jiang
Account Name: chenfeiw

1. Introduction

This competition requires training and testing on the data sets available on Kaggle. This dataset contains student information collected from 7 different districts in the United States, including 50 SRP (Student Retention Predictor) scores, student self-evaluations scores, teacher evaluations scores, student extracurricular performance scores, and district coding. Our goal is to predict students' standardized test scores y.

In this competition, we first analyzed the features in the dataset. After preprocessing, we tried to use different models to train and predict the data. After comparative analysis, we found that the neural network model has the best prediction effect. Therefore, we used PyTorch to build a neural network and optimized the parameters of the model, and the final R^2 reached 0.78984.

2. Data

There are 8,000 pieces of data in the training set and 4,000 pieces of data in the test set. The rest of the features include:

1. Student self-evaluation ranges from 1-5.
2. The teacher's evaluation of the student ranges from 1-5.
3. Students' performance in extracurricular activities is rated on a scale of 1-10.
4. The student's school district, coded from 1 to 7.
5. 50 SRP (Student Retention Predictor) scores for the student, each score feature presents a normal distribution, and the range is mostly between -5 and 5.

3. Methods and Results

(1) Data preprocessing

After analyzing the characteristics of the data set, we first perform data preprocessing operations.

We observed that the features in the dataset can be divided into two parts: Numeric features: including 'self_eval', 'teacher_eval', 'extracurricular' and 50 features starting with 'SPR'. Non-numeric (categoric) features: only including 'district'.

For numeric features, we observed that each feature represents a certain score respectively, but the range of scores is different. Therefore, we decided to use StandardScaler() to standardize these variables to facilitate subsequent model training.

For non-numeric (categoric) features, there is only one feature 'district', which represents the district code where the student received education. The number does not represent the order but the difference in categories. Therefore, we use OneHotEncoder() to re-encode this variable and convert it into 7 groups of 0-1 variables to facilitate subsequent model training.

In particular, for the 50 features starting with 'SRP', we attempted to use the PCA method. We noticed that after setting the parameter 'n_components=0.95', there were still 48 features left, which is a slight reduction in dimensionality. This indicates the data cannot be well separated linearly. Therefore, we decided not to use PCA on these features but standardize them along with several features representing scores. We saved the two preprocessing results separately for model training comparison.

(2) Model training and predicting

1. Model exploration and selection

After completing the data preprocessing, we selected several machine learning models for training and prediction. We use R^2 to evaluate the performance of the model.

We first used the train_test_split function to divide our dataset into training and testing sets, and selected linear regression models, gradient boosting regression models, random forest models, neural network models (MLPRegressor) and other models to train the preprocessed data respectively.

We found that the preprocessing method that does not use the PCA method to process the SRP features, but directly standardizes them, is more accurate on every model. Therefore, we abandoned the PCA method and adopted a second preprocessor instead.

We fitted these models on the split training and testing sets, and the scores are as presented in the following table. Since the R^2 of the neural network is significantly higher than other models, we chose to use the neural network model and studied how to optimize its parameters.

Table1. Result of model fit

	Train R2	Test R2
(1) Linear	0.6421	0.6513
(2) Gradient Boosting	0.7044	0.5853
(3) Random Forest	0.9440	0.6061
(4) Neural Network	**0.9879**	**0.7368**

2. Neural network optimization

We used PyTorch to define a neural network model, including 4 linear layers and 3 ReLU activation function layers. The input layer maps 60 input features (one for each feature of the dataset) to 128 features in the next layer. Subsequent linear layers gradually reduce the dimensionality from 128 to 64, then to 32, and finally map the 32 features to a single output value that represents the prediction score.

In all the hidden layers, we used a ReLU (rectified linear unit) activation function, which introduces nonlinearity into the model, allowing it to learn more complex patterns in the data.

We chose to use the Adam optimizer to adjust the model weights. When adjusting parameters, we tried to explore the performance differences of different learning rates, and finally adjusted the learning rate to 0.001 to achieve the optimal prediction effect.

We used the trained neural network model to predict the test set, uploaded the prediction results to the Kaggle website, and obtained a score of 0.78984.

4. Contribution

Yuxiang Gao: Assisted in the modification of the report
Chenfei Wang: Responsible for the overall construction, training and testing of the model
Tongyao Jiang: Proposed the method of neural network