UNIVERSITY OF MICHIGAN

STATS415 Data Mining and Statistical Learning

Kaggle Report

Project Group 3
Runhui Xu
Ziyang Xiong
Xixiao Pan
Huijie Tang

Kaggle Account Name: RunhuiXu

December 3, 2023

1 Data Preprocessing

1.1 Train and Test Data Split

To evaluate our model and adjust the hyper-parameters in the model, we first split 80% of our training samples into new training data set and the rest as test data set. We set the random seed to 1 and use the sample function in R to draw new training and test data set.

1.2 Self Evaluation, Teacher Evaluation and District

The data set has first provided us with self_eval – the student's evaluation of their own mastery of the content, teacher_eval – the student's teacher's evaluation of the student's mastery of the content and district – which school district the student came from. Self evaluation and teacher evaluation both have integer value from 1 to 5 and district has integer value from 1 to 7. It is important to determine whether they are numeric or categorical value. To dig deeper into this, the following boxplot is made in Figure 1.

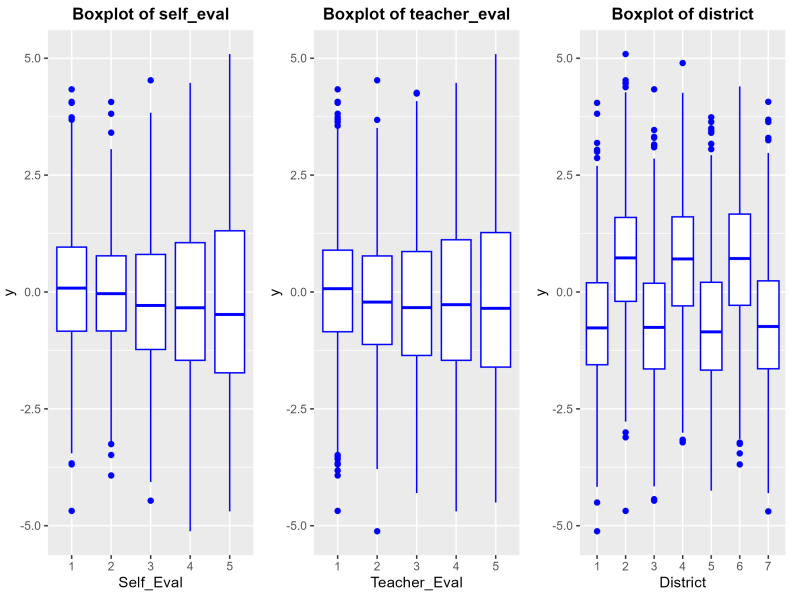

Figure 1: Boxplots between y and three predictors.

From the observations depicted in Figure 1, it is evident that the mean value of y exhibits a subtle decreasing trend with the increase in self_eval and teacher_eval. Consequently, it is reasonable to treat these variables as numeric values rather than categorical classes. Intuitively, one would expect that both self-evaluation and teacher evaluation, as numeric measures, would exert some influence on the final score. This intuition has been validated through practical experimentation, as treating these predictors as numeric values yielded superior results on the test dataset. To make sure that all numeric data have the same basic significance, we scaled them to have mean 0 and variance 1.

In the case of the district predictor, no discernible consistent increasing or decreasing trend was identified. However, there appears to be a distinction in mean values of y between even-numbered (2, 4, 6) and odd-numbered (1, 3, 5, 7) districts. As a result, we

introduced a dummy variable to signify whether a sample originates from an even or odd school district.

1.3 SRP Data

According to the plot of 50 days' measurement of the SRP component, we observe no apparent distributions, but it looks like a superposition of a series of wave functions. Inspired by Anomal, as a time series data, the line plot for 50-day SRP data shows the non-stationarity in Figure 2. To enhance the clarity of the data representation, the Fourier Transform is applied to transform time-domain data into frequency-domain data as shown in Figure 3. Every sample has its unique transformation. Then, we represent the original time function with different frequency components. The higher spectral density indicates greater effectiveness of the corresponding frequency component. Consequently, we identify the top three peaks in the frequency spectrum for each sample and record the respective frequency. The process is applied to every sample, and finally we replace the 50 days SRP data with three top1, top2 and top3 frequency variables across all datasets. To simulate the period of the wave functions, the three $\frac{1}{freq}$ values are also added into the input.

However, in the process of characterizing frequency, we lose the influence of amplitude in the time-domain figure. Thus, we also add the variance of the 50 days' measurement of every sample as a new variable, which represents the amplitude. The mean of the 50-day SRP data is also added.

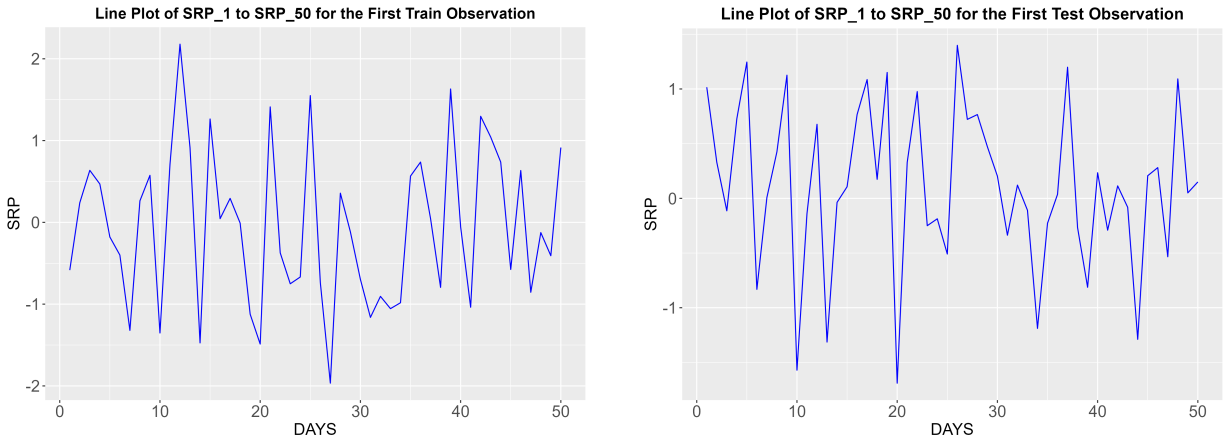

Figure 2: Line Plot for 50-day SRP Data of First Train and First Test Sample

2 Model

The final data after preprocessing include 3 estimated periods and 3 frequencies for SRP data, mean and variance of SRP data, scaled self_evaluation and teacher_evaluation and a dummy variable for district. In the data processing phase, we identified a non-linear relationship between predictors and the response variable in our regression problem. To address this, we opted for boosting, an enhanced version of decision trees, and leveraged the xgboost package in R. Since xgboost requires a numeric matrix as input, we treated

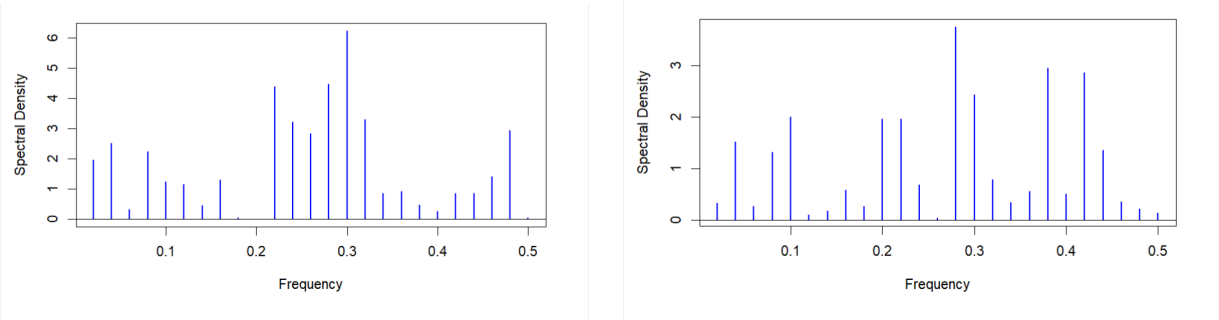

Figure 3: Periodgram for First Train and First Test Sample

the district predictor as a dummy variable instead of a categorical value. The transformation of the training data frame into matrix format was achieved using the as.matrix and xgb.DMatrix functions before feeding it into the model.

Our focus during model tuning centered on adjusting key hyperparameters such as the number of rounds (nrounds), maximum tree depth (max_depth), and the learning rate (eta). Employing the caret package's random search method for hyperparameter tuning, we utilized xgb_model$besttune to obtain the parameters of the best-performing model based on either bootstrap or cross-validation. Subsequently, we manually fine-tuned the model by iterating through possible values of hyperparameters to assess their impact on test dataset performance. Ultimately, we selected the best model, characterized by the optimal set of parameters.

Post-hyperparameter adjustment, we trained the model on the entire dataset. To mitigate randomness in the final outcome, we employed bootstrapping during the model fitting process. Given our 8000 training data samples, we generated 8000 samples with replacement (replace == TRUE) using the sample function. The final model was fitted 1000 times, and the average value of these iterations was computed to derive the conclusive result.

3 Contribution

All members tried different methods (including SVM, KNN, Tree) to build the model and decided to use boosting after discussing together. Runhui Xu contributed to data pre-processing and visualization, and was mainly responsible for Kaggle submissions. All members evenly contributed to improving the model and report writing.