Forecasting students' future academic performance using big data analytics

Zhen Li, Steven Tang

eMetric LLC

Paper written for the 2019 meeting of the National Council on Measurement in Education, Toronto, Canada. The views expressed in this paper are solely those of the authors and they do not necessarily reflect the positions of eMetric LLC.

Correspondence concerning this paper should be addressed to Zhen Li, eMetric, 211 N Loop 1604 E, Suite 170, TX 78232. Email: zli@emetric.net.

Abstract

In education, big data analytics methods have become increasingly popular (Romero & Ventura, 2010). This article illustrates how we use XGBoost regression trees for predicting students' future performance in state summative tests. Bayesian networks and linear regression model are applied for comparison. Results show that XGBoost regression trees perform the best, with higher prediction accuracy and computation efficiency. The XGBoost regression tree also works better with incomplete data sets.

Keywords: XGBoost regression tree, Bayesian networks, *K-12* assessment

Year after year, students take high stakes summative tests, and the results of these tests can have far-reaching consequences for students, teachers, and other stakeholders. In this study, we investigate the possibility of using the XGBoost statistical framework, which implements gradient boosted regression trees, in order to make potentially useful forecasts of student scores on high stakes summative tests. Given the current and prior scores of a particular student, we seek to forecast how that student will do on next year's tests. This type of information could be useful to many stakeholders; teachers and schools could draft a plan to create targeted interventions for at-risk students, for example. The underlying hypothesis is that modern methods such as XGBoost regression have proven to be statistically accurate and operationally easy to use and may be able to provide a feasible statistical framework to provide score forecasts, and such predictions could eventually be disseminated via reporting to various stakeholders. We seek to compare XGBoost results to other commonly used statistical frameworks in education literature, namely Bayesian networks and linear regression. The statistical frameworks will be evaluated using overall predictive accuracy (root-mean -square error) as well as robustness to missing data.

The Big Data Analytics Models

XGBoost regression tree (XGBoost). This approach relies on iteratively building a collection of simple regression trees; regression trees are decision trees that predict continuous outcomes (Chen & Guestrin, 2016). The iterative process starts by first creating an extremely simple predictive regression tree; such a tree might only have between 2 to 16 leaf nodes. This initial regression tree is constructed by searching through a large number of potential split values among all input variables and finding the splits that minimize prediction error. The iterative process continues by constructing an additional regression tree of the same structure, but this time constructed to minimize the *residual errors* of the first regression tree. The next iterative tree is then constructed to minimize the residuals of the full model thus far, and the process of iteratively creating new trees continues until stopping criteria is met. As the name implies, gradient boosting uses gradient descent to find the next regression tree to add to the ensemble. At the end of the building process, the predictions are given by the sum of the outputs of all trees. This process of building a gradient boosted regression tree was optimized in the XGBoost package allowing for very fast computation of gradient boosted trees as well as many opportunities for additional model tuning (Benjamin, Fernandes, Tomlinson, Ramkumar, VerSteeg, Miller, & Kording, 2014).

For a predictive model $\hat{y}_1 = f_1(X)$, where X indicates input variables, $\hat{y}_1$ indicates predications by the first tree and y indicates the observed output variable, a loss function can be defined between the prediction and the observed outcome: $l(\hat{y}_1, y)$. During training, the first tree can be estimated by minimizing the following objective:

$$L_1 = \sum l(\hat{y}_1, y) + \Omega(f_1) \tag{1}$$

Ω is a regularizing function to avoid overfitting. Then a second tree $f_2(X)$ will be constructed by predicting the residuals of the first tree. The objective to minimize is as follows:

$$L_2 = \sum l(\hat{y}_1 + f_2(X), y) + \Omega(f_2) \tag{2}$$

The process continued sequentially for a fixed number of trees (*N*). Total loss will be progressively decreased with each additional tree. In the end, the prediction for y will be the sum of the predictions of all trees:

$$\hat{y} = \sum_{k}^{N} f_k(X) \tag{3}$$

Compared to linear regression and quantile regression, XGBoost regression tree require completely different assumptions. For example, linear regression has a basic assumption that the sum of its residuals is 0. XGBoost regression tree, through its boosting process, instead attempts to find and model patterns in the residuals and strengthen the model with weak learners that exploit these patterns. This approach has shown to be extremely powerful in big data tasks, winning a variety of competitions where predictions need to be made based on a wide set of predictors.

Bayesian networks (BN). Based upon a joint distribution for a directed acyclic graph, Bayesian networks can estimate conditional probability of one variable given other variables in the net. As we know, building a Bayesian net consists of two parts: structure learning and parameter learning. The structure of a net can be either freely estimated or pre-defined. In this study, we compared results from a learned structure and a fixed structure and found the prediction results very close to each other. With a large number of input variables, structure learning is very time demanding. Therefore, a simple fixed structure was applied for all the Bayesian networks modeling.

$$P(y|\boldsymbol{X}) = P(y)\prod_{k=1}^{n} P(x_k|y) \quad (1)$$

Where $\boldsymbol{X} = (x_1, \dots, x_k, \dots, x_n)$ indicates the input variables, y indicates the score field to be predicted. The number if input variables is n. The net only has edges from all the input variables to the target variables, which means that the target variable is dependent on all the input variables. Furthermore, all the input variables are assumed to be independent. The parameters of the structure (conditional probabilities) were freely estimated by maximum likelihood estimation. The R package "bnlearn" is used for parameter calibration (Scutari, 2010). As all functions in “bnlearn” require complete data, the training data only contains students with complete observations. For the test data, we impute the input variables with the learned net at the first step and predict the target variables at the second step.

Bayesian networks (Pearl & Scutari, 2000; Scutari, 2010) have been thoroughly studied for several decades and is also popular in the psychometrics field (Pearl & Scutari, 2000; Mislevy, Almond, Yan & Steinberg, 2000; Tsamardinos, Brown, & Aliferis, 2006; Sinharay, 2006; Scanagatta, de Campos, Corani, & Zaffalon, 2015). Comparing to other machine learning models, Bayesian networks have shown several advantages. First, expert knowledge of the net structure and conditional probabilities can be incorporated. Second, all the parameters in Bayesian networks are interpretable and can be presented clearly in a graph. Third, no specific input and output variables need to be defined. That is to say, once the net is learned and calibrated, the values of any variable can be predicted using the other variables. Fourth, Bayesian networks have also been found to be robust to missing data (Friedman, 1997). Fifth, likelihoods can be provided to predicted scores. Finally, Bayesian networks have been applied in psychometrics for decades. For example, Mislevy et al. (2000) applied Bayesian networks to model relationships between latent cognitive variables; Sinharay (2006) applied the posterior predictive model checking method to evaluate model fit of Bayesian nets. Therefore, we select Bayesian networks as our second method.

Methodology

Data

One cohort of students' test scores in reading, writing, math, and science from grade 3 to grade 8 were collected. Science was only taken in grade 5 and grade 8. The following table shows the subjects tested at each grade.

Table 1.
Test Data per Grade

	Grade 3	Grade 4	Grade 5	Grade 6	Grade 7	Grade 8
Reading	√	√	√	√	√	√
Math	√	√	√	√	√	√
Science			√			√

Note: "√" means that the subject was tested at the purported grade.

Test scores included scale scores, performance levels, as well as reporting category scores for each subject. About a quarter of students had incomplete records. Additionally, students' demographic information, e.g., gender, ethnicity, were also included in the data input file. In the output variable (predicted field), only valid test scores were selected. The total number of students in each test ranged from 300,000 to 400,000. 80% of the data was randomly chosen for training and validation, while the remaining 20% was used as a test dataset.

Study Design

The aim of this study is to evaluate XGB in predicting students' next-year academic performance in summative tests. We compare XGB with two popular approaches: Bayesian networks and linear regression. In the prediction model framework, the input variables include all previous years' test scores and students' demographic information (2013-2017). The output variables are test scores at the most recent year (2018). For students in a lower grade, e.g., grade 4, only one previous-year data exist (e.g., grade 3 in 2017); However, students in a higher grade, e.g., grade 8, have many more previous years of test data (e.g., grade 3 in 2013 - grade 7 in 2017). In this study, we also explore how the prediction accuracy of XGB could improve when more previous years of test data are used as input variables. In the end, we compare the performance of XGB and Bayesian networks with regard to their prediction accuracy for students with incomplete data.

Evaluation Criteria

We used root mean squared error (RMSE), mean errors (ME) and classification consistency to evaluate the performance of the prediction models.

$$RMSE = \sqrt{\sum_{i=1}^{N}(SS_{forecast} - SS_{observed})^2 \Big/ N}, \qquad (2)$$

$$ME = \sum_{i=1}^{N} (SS_{forecast} - SS_{observed}) \Big/ N, \tag{3}$$

where N is the total number of students for a test; $SS_{forcast}$ indicates predicted scale scores; $SS_{observed}$ indicates the observed scale scores.

Classification consistency is defined as the probability that the predicted scores and real scores classify students into the same performance level group, based on the given performance level cuts for each test.

Results

The three above-mentioned methods for predicting students' academic performance were applied to a longitudinal data set, consisting of students' test scores for 6 years in a state assessment. We predicted students' scale scores of different subjects at Grades 4-8 by all their corresponding previous-year data. Results are presented in this section.

Model fit

Psychometric models commonly report one or several model fit indices when applied to real data. However, machine learning packages do not produce model fit indices directly. Usually, machine learning models are evaluated using different training, validation and test datasets. The prediction errors on the validation and test data set are the major criteria of evaluation. XGBoost also produce the loss functions across training. Figure 1 shows an example of the training and validation loss function across iterations by XGBoost regression tree. Prediction errors for the training and validation data decrease at the same time with more iterations, which provides evidence that overfitting doesn’t happen. More complex model evaluation, such as cross validation, could also be carried out for both methods. But as our sample size is very large while the number of input variables is relatively small, it is evident that the training, validation and test data in our study are all representative of the full data.

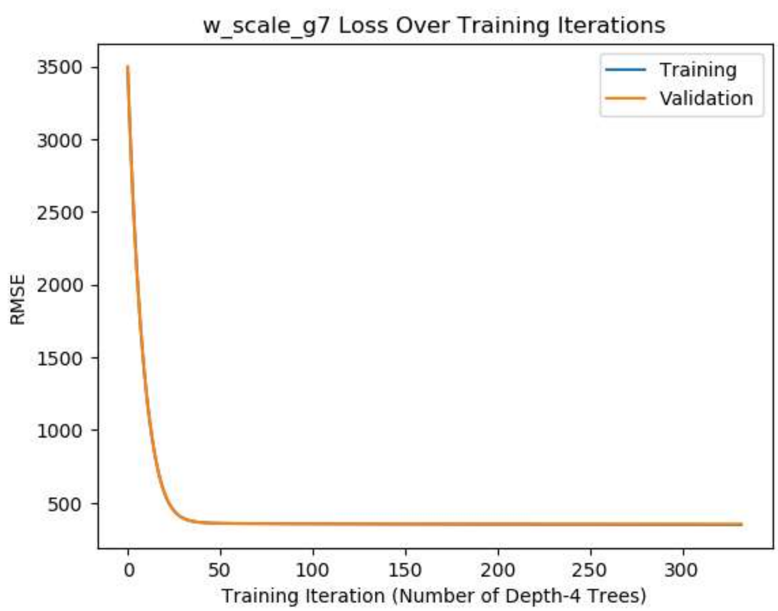

Figure 1 Loss over training iterations by XGBoost

Classification Consistency

Using the predicted scores, classification consistency indices were calculated based on known cut-off scores. From 2012-17, this test has two fixed cut-off standards: “Performance Level Cut 1” and “Performance Level cut 2”. Table 2 presents the classification consistency at each performance level cut respectively.

Table 2

Comparison of classification consistency index for two performance level cuts

	Performance Level Cut 1			Performance Level Cut 2		
Target Field	Linear regression	Bayesian networks	Regression tree	Linear regression	Bayesian networks	Regression tree
G8 Math	0.698	0.742	**0.788**	0.882	0.896	**0.901**
G8 Reading	0.822	0.815	**0.845**	0.858	0.863	**0.874**
G8 Science	0.802	0.804	**0.818**	0.856	0.878	**0.885**
G7 Math	0.832	0.839	**0.853**	0.899	0.903	**0.909**
G7 Reading	0.820	0.819	**0.842**	0.856	0.866	**0.876**
G6 Math	0.780	0.831	**0.845**	0.882	0.910	**0.915**
G6 Reading	0.786	0.832	**0.847**	0.846	0.884	**0.889**
G5 Math	0.784	0.818	**0.822**	0.863	0.882	**0.885**
G5 Reading	0.787	0.828	**0.833**	0.853	0.877	**0.880**
G5 Science	0.759	0.808	**0.810**	0.898	0.910	**0.911**
G4 Math	0.797	0.823	**0.826**	0.857	0.884	**0.885**
G4 Reading	0.803	0.820	**0.834**	0.830	**0.871**	**0.871**

Table 2 shows that classification consistencies for the predicted scale scores by XGBoost are higher in all conditions. Mostly, the classification consistencies for the predicted scale scores by Bayesian networks are close to those by XGBoost regression tree, and much higher than those by linear regression. One exception is for Grade 8 reading test, the classification consistency index for the predicted score by Bayesian networks at the first performance level cut standard is lower than that by the linear regression.

Prediction Errors

The precision of predicted scores by three models was further evaluated using RMSE. Figure 2 presents RMSE results of three methods.

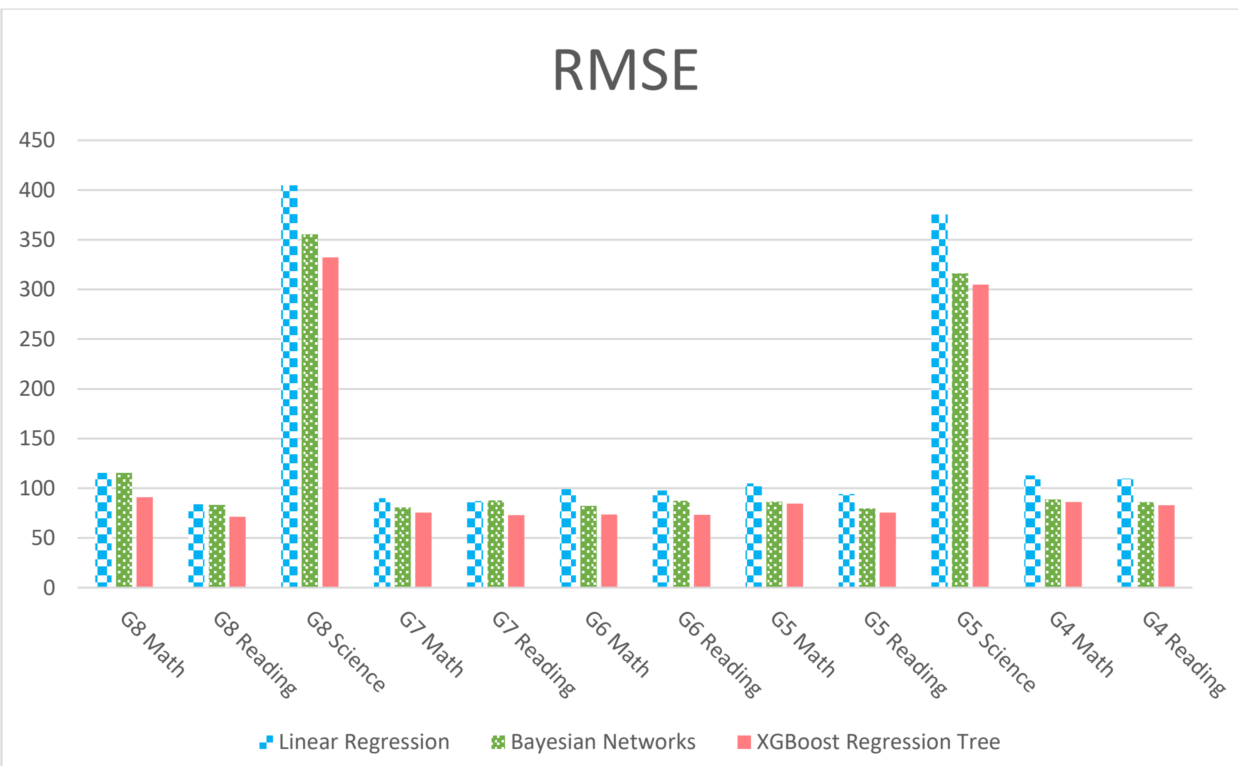

Figure 2 RMSE for all tests by three methods

Figure 2 shows that the XGBoost regression tree has the smallest RMSE among the three methods. Bayesian networks are slightly worse than XGBoost and better than linear regression for most subjects and grades, except for grade 8 mathematics. In addition, we also compute the mean errors and find that XGBoost has the most stable and lowest mean absolute errors across all tests (see Figure 3).

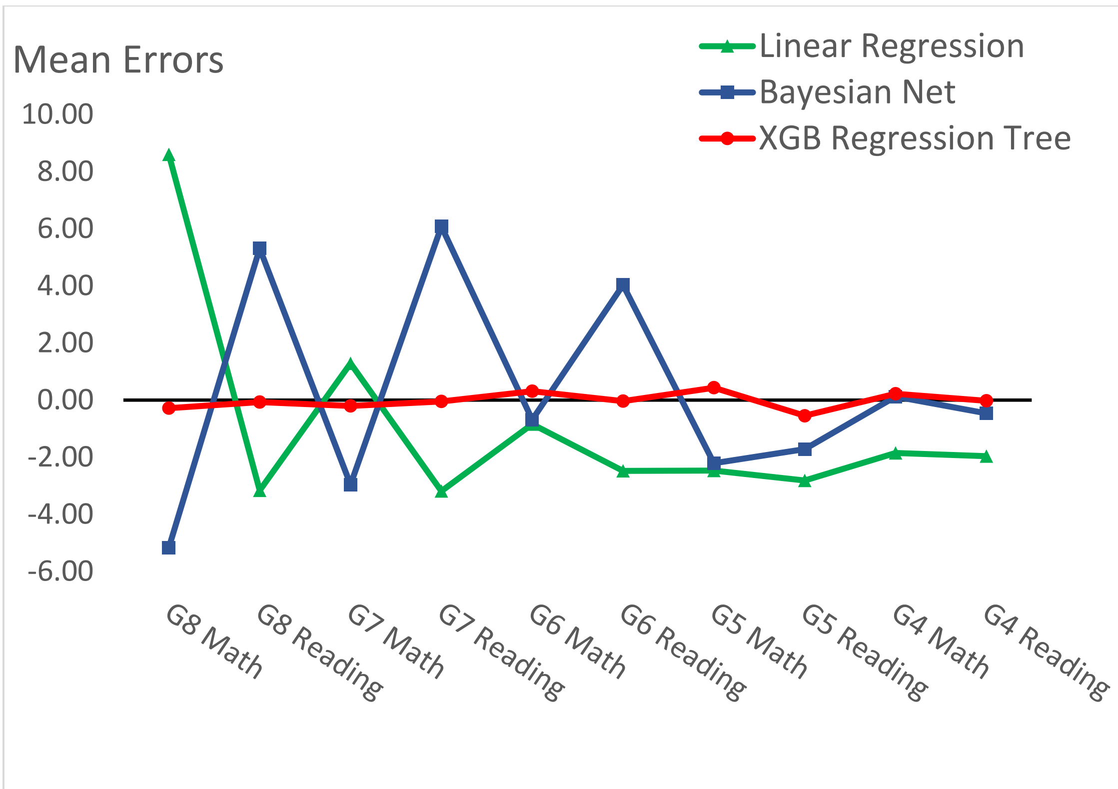

Figure 3 Mean absolute errors for all tests by three methods

The reason why Bayesian networks don't perform well might be that quite a few students have missing values for their previous-year scores, and Bayesian networks would provide bad predictions for these students. On the contrary, XGBoost provides better predictions for students with missing data. In the next section, we conducted some further analysis to test our hypotheses.

The Prediction for Students with Incomplete Data

Generally speaking, students with incomplete inputs have less accurate predicted scores than the students who have complete input variables. Among the three methods, XGBoost regression trees can handle missing data the best, with the highest efficiency. It is able to train models with incomplete datasets and make predictions for incomplete data; The trained model remains stable with or without missing values. Table 3 shows the RMSE for complete and incomplete test datasets respectively, when the XGBoost model was trained with both complete and incomplete data.

As a comparison, incomplete data needs to be attended more carefully in Bayesian networks modeling. First, as mentioned above, all functions in 'bnlearn' requires complete data, thus only students with complete data are included in the training data set; Second, variables with only one constant value are removed from the inputs, otherwise parameters will contain zeros and predictions cannot be generated; Third, for students with incomplete data in the test dataset, imputation needs to be carried out for all students to get a prediction; Fourth, when the number of input variables is large (e.g., 117 input variables for Grade 6), the structure learning process becomes extremely computationally demanding. This was one of the reasons why the net structure was fixed in our study, which might not be the best model for imputation and prediction. Nonetheless, as shown in Table 4, with all the above issues considered, Bayesian networks can provide adequate predicted scale scores. The model is also very stable with incomplete data. The existence of incomplete data doesn’t exert an influence on the prediction of students with complete data.

Table 3

RMSE for students with complete or incomplete data using XGBoost

Target Field		Complete		Incomplete	
	N_Train	N_Test	RMSE	N_Test	RMSE
Grade 8 Math	259282	42506	78.4	22315	112.3
Grade 8 Reading	304416	57770	66.7	18335	84.3
Grade 7 Math	263172	52388	67.7	13405	101.0
Grade 7 Reading	290297	58034	67.1	14541	92.9
Grade 6 Math	279032	59056	69.4	10702	93.3
Grade 6 Reading	286567	59875	67.3	11767	98.0
Grade 5 Math	287105	63300	80.2	8477	110.8
Grade 5 Reading	288978	63815	70.6	8430	105.9
Grade 4 Math	287388	67590	82.9	4258	128.5
Grade 4 Reading	287653	67401	78.4	4513	132.5

Table 4

RMSE for students with complete or incomplete data using Bayesian Networks

Target Field		Complete		Incomplete	
	N_Train	N_Test	RMSE	N_Test	RMSE

Grade 8 Math	136283	42722	79.2	22099	164.6
Grade 8 Reading	185521	58041	66.7	18064	121.7
Grade 7 Math	168379	52624	68.2	13169	118.8
Grade 7 Reading	187311	58334	67.4	14241	143.5
Grade 6 Math	189761	59329	70.9	10429	129.6
Grade 6 Reading	192992	60187	67.6	11455	154.6
Grade 5 Math	204358	63650	80.6	8127	122.9
Grade 5 Reading	205342	64130	70.9	8115	129.1
Grade 4 Math	217450	67965	83.9	3883	153.4
Grade 4 Reading	217133	67816	79.5	4098	160.0

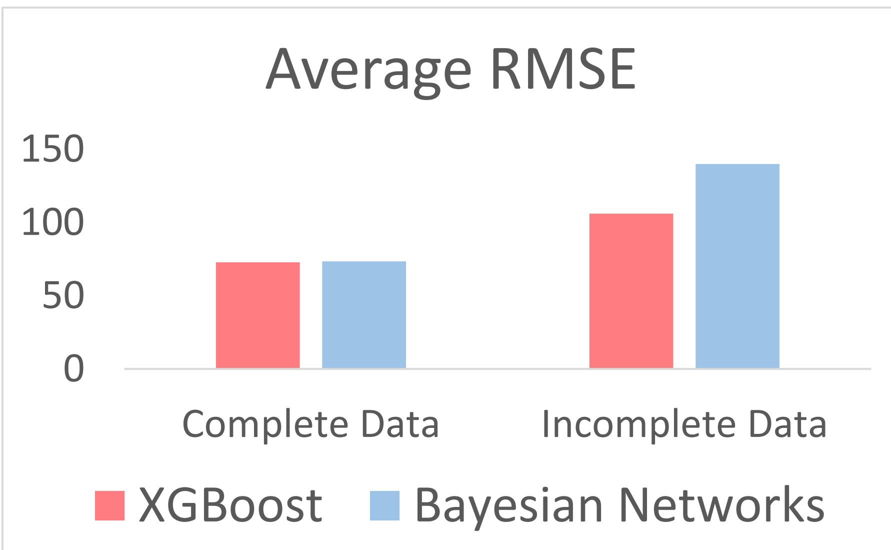

Figure 4 The average RMSE across grades and subjects

As shown in Figure 4, students with incomplete inputs have less accurately predicted scores than the students who have complete input variables. XGBoost handles missing data innately. Specifically, it is able to train models with incomplete datasets;

the trained model remains stable with or without missing values. As a comparison, Bayesian networks provide less accurate predicted scale scores for students with incomplete data, even though the missing values were attended more carefully. Nonetheless, the existence of incomplete data doesn't exert an influence on the prediction of students with complete data for both methods.

How Many Previous Years of Data Are Needed?

The prediction errors of XGBoost regression trees using different number of previous-year scale scores are computed. Figure 5 shows that when the number of previous years increased, the prediction accuracy also increased.

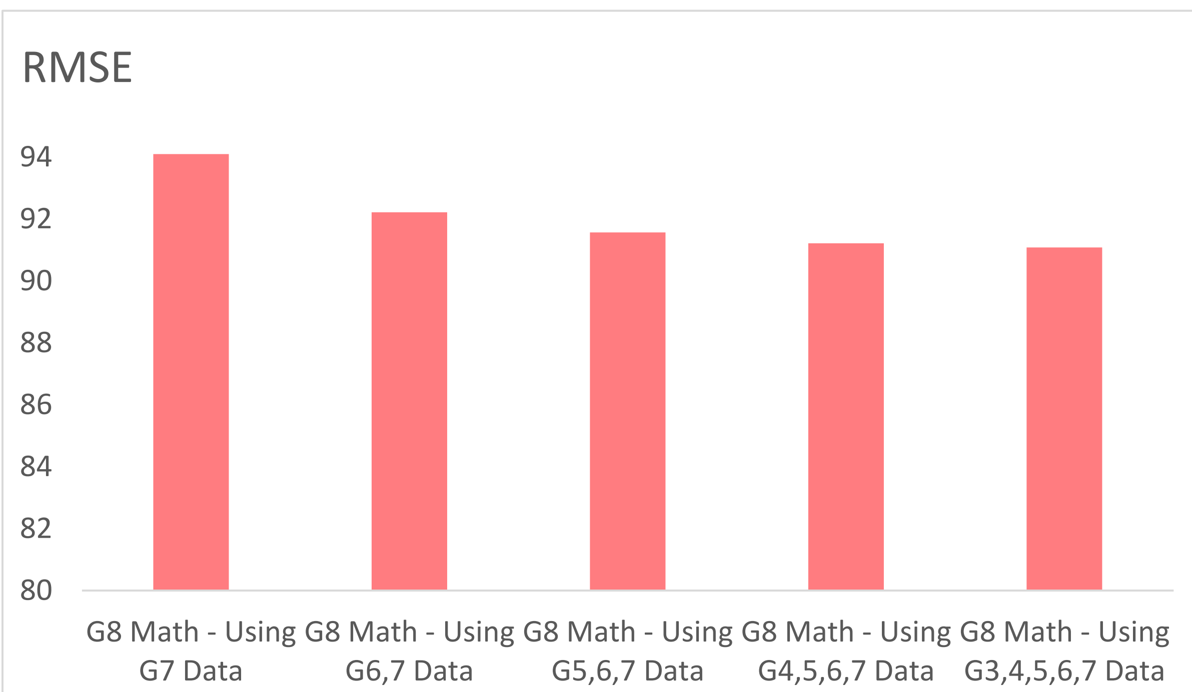

Figure 5 Decreasing prediction errors with more previous years of data

Conclusion

The practical purpose of this research is to investigate the practicality of using a statistical framework like XGBoost to forecast scores for next year's tests. The hope is that forecasted scores could then be acted upon by stakeholders, perhaps to identify areas of weakness or focus on at-risk students. In this study, we only predicted future overall scale scores, but the XGBoost statistical framework should be capable of predicting other more specific outcomes, such as more specific test subjects (known as reporting categories in many states).

The results indicate that among the 3 statistical approaches (XGBoost, Bayesian Networks, Linear Regression), XGBoost had the best predictive accuracy. This can be expected given the expressive and robust nature of XGBoost, which has proven itself across many big data predictive tasks. In this study, we tuned the XGBoost algorithm specifically for longitudinal test data and were able to successfully create accurate forecasted results. Operationally, XGBoost is very easy to use, as it handles data with missing and incomplete values inherently. Unlike other big data methods, XGBoost offers good interpretive properties as well, enumerating exactly how the model arrives at its output. On the contrary, Bayesian networks require additional considerations in handling missing data, and provide less accurate predictions for students with incomplete data.

There are many possible statistical frameworks that could underly models that forecast future performance, and there are almost certainly many additional refinements we could have made to the Bayesian Networks and linear regression models in this study. Our overarching hypothesis, though, is that methods like XGBoost will be able to provide the most accurate predictions even as the number of explanatory variables expand, as expressive models like XGBoost have shown to be very successful across many big data prediction tasks. The results presented in this study can contribute to a fuller understanding of how modern statistical methods can solve or improve on problems of prediction in large-scale measurement.

Reference

Benjamin, A.S., Fernandes, H.L., Tomlinson, T., Ramkumar, P., VerSteeg, C., Miller, L., & Kording, K.P. (2018). Modern machine learning far outperforms GLMs at predicting spikes. *Frontiers in Computational Neuroscience, 12 (56), 1-13.*

Chen, T. & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System.* Paper presented in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco.

Friedman, N. (1997). *Learning Belief Networks in the Presence of Missing Values and Hidden Variables.* Paper presented in Proceedings of the 14th International Conference on Machine Learning, San Francisco, pp. 125–133.

Pearl, J., & Russell, S. (2000). Bayesian networks. [online]. Retrieved from https://escholarship.org/uc/item/53n4f34m

Mislevy R. J., Almond R. G., Yan D., Steinberg L. S. (2000). *Bayes nets in educational assessment: Where do the numbers come from?* (Tech. Rep. No. 518). Los Angeles, CA: National Center for Research on Evaluation, Standards, and Student Testing.

Romero, C. & Ventura, S. (2010). Educational data mining: a review of the state-of-the-art. *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews)*, *40(6):*601-618.

Scanagatta, M., de Campos, C. P., Corani, G., & Zaffalon, M. (2015). *Learning Bayesian networks with thousands of variables.* Paper presented at 29th Conference on Neural Information Processing Systems, Montreal, Canada.

Scutari, M. (2010). Learning Bayesian Networks with the bnlearn R package. *Journal of Statistical Software*, *35(3):* 1-22.

Sinharay, S. (2006). Model diagnostics for Bayesian networks. *Journal of Educational and Behavioral Statistics, 31,* 1-33.

Tsamardinos, I., Laura E. B., & Constantin F. A. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning, 65(1):* 31-78.