Educational Measurement
ISSUES AND PRACTICE

NCME national council on measurement in education

Educational Measurement: Issues and Practice
Fall 2016, Vol. 35, No. 3, pp. 38–54

An NCME Instructional Module on Data Mining Methods for Classification and Regression

Sandip Sinharay, *Pacific Metrics Corporation*

Data mining methods for classification and regression are becoming increasingly popular in various scientific fields. However, these methods have not been explored much in educational measurement. This module first provides a review, which should be accessible to a wide audience in education measurement, of some of these methods. The module then demonstrates using three real-data examples that these methods may lead to an improvement over traditionally used methods such as linear and logistic regression in educational measurement.

Keywords: bagging, boosting, classification and regression tree, random forests

Measurement practitioners are often interested in problems in which the main goal is the prediction of a response variable from several predictor variables. Such problems belong to one of two categories: *regression problems* where the response variable is continuous (such as the score on a test), and *classification problems* where the response variable is discrete (such as the pass/fail status on a test) and is often referred to as a *class*.

Regression problems in educational measurement include the prediction of estimated item parameters from item attributes (Gorin & Embretson, 2006; Sheehan & Mislevy, 1994), predictive validity studies (e.g., Kuncel, Wee, Serafin, & Hezlett, 2010), prediction of item difficulty from item position (Li, Cohen, & Shen, 2012), and prediction of one test score from another test score (e.g., Moses, 2012). Classification problems in educational measurement include electronic essay scoring in which an integer essay score (between, e.g., 1 and 6) is predicted from several essay features (e.g., Ramineni & Williamson, 2013), prediction of pass/fail outcomes in high-stakes testing (e.g., Schmidt, 2000), course placement (e.g., Schulz, Betebenner, & Ahn, 2004), prediction of whether a student finishes college or doctoral studies from several covariates (e.g., Nettles & Millett, 2006), and prediction of high school graduation and high school dropout from several examinee-level and school-level covariates (e.g., Burrus & Roberts, 2012; Subedi & Howard, 2013).

Most of the above problems are addressed using traditional prediction methods such as multiple linear regression (MLR) and logistic regression (LoR) that are well known, easy to fit, and well researched. However, because these methods involve the assumption of linearity of the relationship between the expected value of the response (or a function of it) and the predictors, they may not always be appropriate and may not be able to explain complex interactions among the predictors (e.g., Draper & Smith, 1998, p. 505; Strobl, Malley, & Tutz, 2009). With increased computational power, one question is whether more computation-intensive prediction methods, which were impractical even a couple of decades ago but are practical now, could lead to better prediction than MLR and LoR in important problems in educational measurement in real time.

Several computation-intensive methods for prediction are available in the field of *data mining* (e.g., Hastie, Tibshirani, & Friedman, 2009), which originated simultaneously in the statistical science and computer science communities. Data mining methods for classification and regression or *methods for supervised learning* are becoming increasingly popular in various scientific fields including social sciences, genetics, epidemiology, medicine, and psychology (see, e.g., Strobl et al., 2009, p. 324, and the references therein). The methods for supervised learning often provide better prediction than traditional prediction methods (e.g., Fernandez-Delgado, Cernadas, Barro, & Amorim, 2014) and can be applied to high-dimensional problems where application of traditional prediction methods is problematic (e.g., Strobl et al., 2009).

In spite of the popularity of the methods for supervised learning in other scientific fields, these methods are yet to be explored in educational measurement with the exception of a few applications of the neural networks to diagnostic classification models (e.g., Gierl, 2007). There exists the "International Educational Data Mining Society" (educationaldatamining.org) with its own journal and own conference. However, the focus of the society is much broader compared to that of the methods for supervised learning (see, e.g., Baker, 2010). Further, Romero and Ventura (2010)[1] stated that

> The educational data mining process *converts raw data* [emphasis added] coming from educational systems into useful information that could potentially have a great impact on educational research and practice.

The focus of the methods for supervised learning is to predict a response variable from the "converted raw data,"

Sandip Sinharay has moved to the Educational Testing Service, 660 Rosedale Road, Princeton, NJ 08541; ssinharay@ets.org.
Note: Any opinions expressed in this publication are those of the author and not necessarily of Pacific Metrics Corporation.

Table 1. Estimated Item Difficulty for the TIMSS Data

	Cognitive Domain			
Topic	**Knowing (Knw)**	**Applying (App)**	**Reasoning (Rsn)**	**Average**
Organizing and representing (O&R)	–	−1.08	−.47	−.77
Number sentences with whole numbers (NSW)	−.34	−.78	–	−.52
Reading and interpreting (R&I)	−.84	−.16	.15	−.40
Points, lines, and angles (PLA)	−.20	.08	–	−.06
Patterns and relationships (P&R)	–	−.13	.28	−.02
Whole numbers (WhN)	−.39	−.00	.79	.05
Two- and three-dimensional shapes (23S)	.09	.18	.15	.14
Fractions and decimals (F&D)	.38	−.32	.96	.38
Average	−.14	−.03	.35	.00

that is, the data that are typically produced by the educational data mining process. For example, in the context of essay scoring, the educational data mining process would extract several numerical features from the essays (using methods such as text mining; e.g., see He & Veldkamp, 2012) while the methods for supervised learning would be used to predict an essay score from those features. As an outcome, there is a lack of research on application of these methods in the educational data mining literature, Antunes (2011) and Abu-Oda and El-Halees (2015) being exceptions(even these exceptions deal with data that are not from large-scale or high-stakes tests).

The purpose of this module is to further explore the use of the methods for supervised learning in educational measurement. After a review of several methods for supervised learning, it is demonstrated using three real-data examples that these methods may lead to some improvement over traditional prediction methods in important prediction problems in educational measurement.

Three Real-Data Examples That Involve Prediction

Predicting Estimated Item Parameters From Item Attributes

Researchers such as Gorin and Embretson (2006) and Sheehan and Mislevy (1994) predicted estimated item difficulty and item slope from item attributes in the context of several large-scale tests such as Praxis^{TM}, Graduate Record Examination$^{®}$ and the National Assessment of Educational Progress.

Let us consider a data set from the Mathematics Assessment for Grade 4 in 2007 of the Trends in International Mathematics and Science Study (TIMSS; Martin & Kelly, 1996). The data set includes the estimated item slope and item difficulty parameters (obtained by fitting the three-parameter logistic model) of 166 dichotomous items. For each item, three item attributes, *topic*, *cognitive domain*, and *reading demand*, are available along with a few other attributes.

Table 1 shows the eight levels of topic, three levels of cognitive domain, their acronyms, and the average estimated difficulty of the items corresponding to each combination of topic and cognitive domain. For example, the average estimated difficulty of the items on the topic O&R and cognitive domain App is –1.08. A blank in a cell of the table indicates that there was no item in the data set corresponding to that combination. The average estimated difficulty for each topic (last column), each cognitive domain (last row), and over all items (last column of the last row) are also shown in the table. The levels of both topic and cognitive domain are sorted with respect to an increasing order of the average estimated difficulty of the items corresponding to each level. Reading demand has three levels: low, medium, and high.

The item attributes were converted to indicator variables. For example, "reading demand" was converted to two indicator variables, one indicating whether the reading demand is medium and another indicating whether the reading demand is high.[2] An MLR model[3] was fitted to predict the estimated item difficulty from these indicator variables. The estimated regression coefficients corresponding to all the seven indicator variables for topic, one indicator variable for cognitive domain, and no indicator variable for reading demand were significant. The MLR model explained 24% of the variability in the estimated item difficulties, which is in agreement with percentages between 20 and 50 reported by Sheehan and Mislevy (1994) and Gorin and Embretson (2006). Later, the performance of several methods for supervised learning would be compared to that of MLR for this data set. Prediction of estimated item difficulty as well as item slope will be considered.

Predicting High School Dropout

Burrus and Roberts (2012) stated that the number of 16- to 24-year-old dropouts in the United States will probably exceed four million at any point of time. They also noted that to keep students in school, one needs to know, as soon as possible, which students are most likely to drop out so that they can be provided with special attention. Accordingly, researchers such as Subedi and Howard (2013) have examined the problem of prediction of drop out from high schools from student covariates. A related question is whether scores on standardized tests are important predictors of student dropout.

Let us consider a data set which includes information on 2,374 at-risk students from 41 high schools, on the potential 2014 high school graduates from the Palm Beach County School District in Florida. A student is *at risk* if she scored in Levels 1 or 2 (out of Levels 1–5) in eighth-grade Reading and Mathematics in the Florida Comprehensive Assessment Test (FCAT). About 16% of the students in the data set actually dropped out. The variables available for each student are as follows:

- Response variable: dropout indicator (i.e., 1 for dropouts and 0 for others).
- Individual-level predictor variables: gender (1 for females and 0 for males), race (1 for African American students and 0 for others), indicator of whether the student is an English language learner (ELL), indicator of whether the student receives free lunch, eighth-grade

Table 2. Estimated LoR Coefficients for the High School Dropout Data

Variable	Estimate	SE	*z*-Value	*p*-Value	Gini
Gender	−.16	.08	−2.12	.03	8
FCAT math score	−.14	.07	−2.03	.04	39
FCAT reading score	−.15	.08	−1.75	.08	42
Average days absent	1.00	.08	12.16	.00	100
Average FCAT score of school	−.22	.20	−1.11	.27	37
Percent retained of school	.75	.12	6.24	.00	45
Percent ELL of school	−.10	.10	−1.02	.31	17

FCAT mathematics and reading scores, average number of days absent per year in high school (AVG_DAYS_ABS), and average number of days of suspension per year in high school.

- School-level predictor variables: average of the eighth-grade FCAT reading and mathematics scores, percentage of students retained (RETENTION_PCT), percent ELL (ELL_PCT), and percent of free lunch recipients of the school the student belongs to.

Subedi and Howard (2013) analyzed a very similar data set to predict high school graduation and dropout using a multilevel LoR model.

A LoR model was fitted to the data set after standardizing the predictor variables. The corresponding estimated LoR coefficients, their standard errors (SEs), z-values (i.e., the estimated coefficient divided by the SE), and two-sided p-values (corresponding to the null hypothesis that the population LoR coefficient is 0) for several predictors are provided in Columns 2–5 of Table 2 (the last column, with title "Gini," would be explained later). Among the 12 predictor variables, the estimated LoR coefficients for four—gender, FCAT Math score, AVG_DAYS_ABS, and RETENTION_PCT (each included in Table 2)—were statistically significant at the 5% level. The correct classification rate (CCR), that is, the fraction of individuals in the data set for whom the predicted[4] and actual values of the response variable were identical, is .86; the corresponding Cohen's kappa (Cohen, 1968) is .42. Later, the performance of several methods for supervised learning for this data set would be compared to that of LoR.

Electronic Essay Scoring

Electronic essay scoring (e.g., Ramineni & Williamson, 2013) is an area of growing interest in educational measurement. Let us consider a data set from an application of electronic essay scoring to a state test, which includes values of several variables computed from the responses of 929 examinees to an essay item. The response variable is a human score, which is an integer between 1 and 4, on the organization of the essay.

The predictor variables are 49 essay features such as measures of errors (such as spelling error and grammatical error) and measures of quality. The traditional prediction methods for electronic essay scoring include MLR of the human score on the features followed by rounding to the nearest integer (e.g., Ramineni & Williamson, 2013) and cumulative LoR (e.g., Haberman & Sinharay, 2010). The CCR and the quadratic weighted kappa (QWK; Cohen, 1968) for the data set were .77 and .75, respectively, from MLR and .80 and .79, respectively, from cumulative LoR. Later, the performance of several methods for supervised learning for this data set would be compared to that of MLR and cumulative LoR.

Data Mining Methods for Classification and Regression

The concept of *test error* is crucial in the context of the methods for supervised learning. Therefore, descriptions of the test error and the closely related cross-validation error are provided first.

Test Error and Cross-Validation Error

In an application of a traditional prediction method, the regression equation is estimated from a sample of individuals. Similarly, in an application of any of the methods for supervised learning, the prediction model corresponding to the method is constructed using a sample referred to as a *training set* or a *training sample*. The methods for supervised learning are not based on probability models. Therefore, closed-form expressions of model-fit criteria (such as SE, fit statistics, Akaike information criterion [AIC], and Bayesian information criterion [BIC]), which are used to evaluate the performance of traditional prediction methods, cannot be computed for these methods. Instead, the performance of the methods for supervised learning is evaluated on the basis of *test error*, which is the expected prediction error over individuals whose data were not used to construct the prediction model (Hastie et al., 2009, p. 220). The test error cannot be computed theoretically for the methods for supervised learning, but can be computed (or estimated) empirically when the values of the predictors and response variable are available for a designated *test set*, which is a sample of individuals that do not belong to the training set and hence were not used to construct the prediction model (e.g., James, Witten, Hastie, & Tibshirani, 2013, p. 30). For each individual in the test set, the predicted value of the response variable is computed using the prediction model and the values of the predictor variables, and the *prediction error* is computed as the discrepancy between the predicted value and the actual value of the individual's response. The test error is the average of these prediction errors over the test set. Usual measures of discrepancy/error are root mean squared error (RMSE) for regression problems and misclassification percentage for classification problems. Thus, for data mining methods for regression problems, the test error would typically be $\sqrt{\sum_j (y_j - \hat{y}_j)^2}$, where y_j and $\hat{y}_j$, respectively, denote the actual and predicted values of the response variable for the jth observation in the test set. Instead of test error, one can compute measures of *test-set accuracy* such as correlation coefficient for regression problems and CCR or Cohen's kappa (Cohen, 1968) for classification problems. It is desirable to have small test error and large test-set accuracy.

While designated test sets are available to researchers (who usually set aside a subset of the available data set as

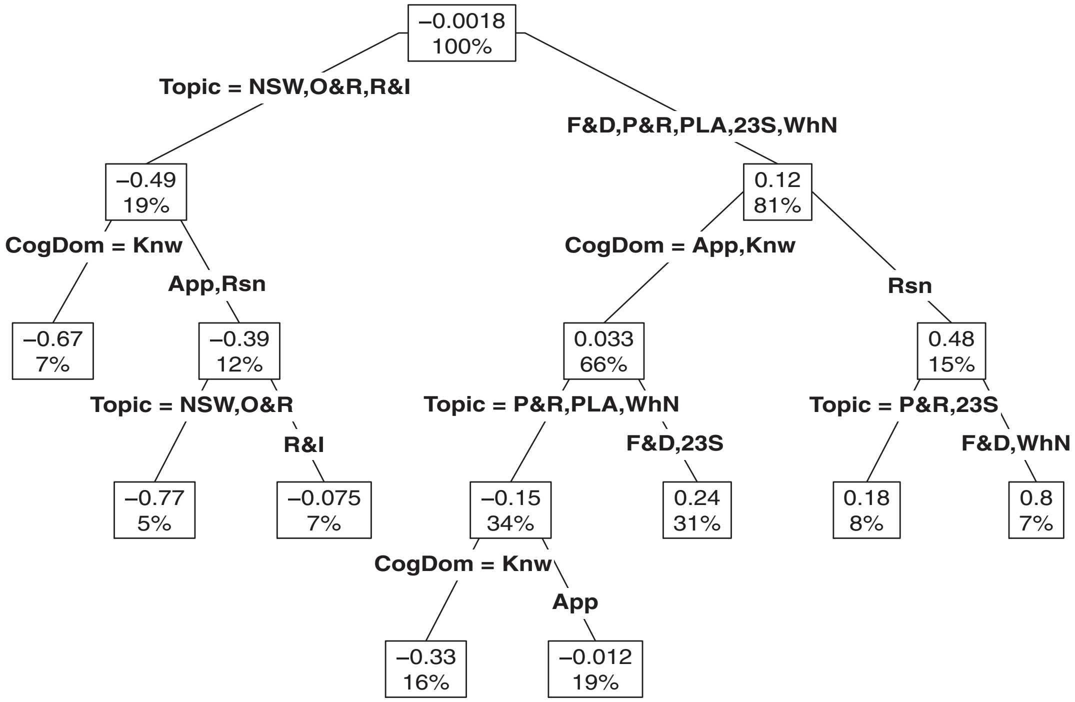

FIGURE 1. A regression tree for predicting item difficulty from item attributes.

the test set), they are usually not available in real-world applications of data mining. Predicted values for the training set mostly lead to an underestimation of the test error[5] (e.g., James et al., 2013, p. 176) and hence should not be used to estimate the test error. In the absence of a test set, one way to estimate the test error is to apply the *K-fold cross-validation* that involves the following steps:

- Partition the training set randomly into K subsets (where K is usually 5–10) of roughly the same size.
- Repeat for $k = 1, 2, \ldots K$ the following steps:
  - Use subsets $1, 2, \ldots k - 1, k + 1, \ldots K$ (i.e., all individuals except those in subset k) to construct the prediction model corresponding to one among the methods for supervised learning.
  - Use the prediction model to predict the responses of the individuals in subset k and compute the corresponding prediction errors.
  - Calculate the average of the prediction errors for subset k.
- Compute the average of the above K average prediction errors.

The average computed in the last step above, referred to as the *cross-validation error*, is similar in concept to test error, in the sense that both of these errors involve prediction for individuals whose data were not used to build the model, and in fact, cross-validation error can be considered as an estimate of the test error (e.g., James et al., 2013, p. 176). To pick the best among several methods for supervised learning in a real application, one can compute the K-fold cross-validation error for each of them and choose the method for which the error is the smallest.

Several methods for supervised learning are extensions of classification and regression trees (CARTs; Breiman, Friedman, Olshen, & Stone, 1984), which are described below.

Classification and Regression Trees

CARTs are nonparametric regression methods in which the prediction formula for predicting the response from the predictors is not explicitly stated and no assumptions are made about the distribution of the variables (e.g., Strobl, 2013). CARTs include *regression trees* that are used in regression problems and *classification trees* (CTs) that are used in classification problems.

Construction of a regression tree. In an application of a CART, the observations in the training set are repeatedly split into two subsets so that the values of the response variable are as similar as possible for all observations within each subset, or equivalently, as different as possible between subsets. For example, a regression tree for predicting the estimated item difficulty, where the above-mentioned TIMSS data set was used as the training set, is provided in Figure 1. The tree was drawn using the R packages *rpart* (Therneau, Atkinson, & Ripley, 2013) and *rpart.plot* (Milborrow, 2011).

In the figure, the predictor cognitive domain is denoted as "CogDom." The several levels for any predictor are denoted using the acronyms introduced in Table 1. In the first step, all the items were split into two subsets, one subset including the items that are on topics NSW, O&R, and R&I constituting the branch on the left and another subset including the items that are on the other five topics constituting the branch on the right. The numbers −0.49 (top) and 19% (bottom) inside the box on the left branch communicate that the average estimated difficulty of the items on topics NSW, O&R and R&I is −0.49 and these items constitute 19% of all items in the data set; note from Table 1 that these three topics correspond to the easiest items on average—so the left and right branches of the tree represent the easy and difficult items, respectively. The right branch includes 81% of all items and their average estimated difficulty is 0.12. The items on the left branch of the tree are further split into two subsets, one including the items on the cognitive domain Knw (to the left; note in Table 1 that the average estimated difficulty for Knw is the smallest among the three levels of cognitive domain) and the other including the items on the two cognitive domains App and Rsn (to the right). The items on the cognitive domain Knw were not split any further. The items on the cognitive domains App and Rsn (to the right) were further split into two subsets, one (average difficulty –0.77) on the topics NSW and O&R (the two easiest topics according to Table 1) and the other (average difficulty –0.075) on the topic R&I. Going back toward the top right of the tree, the items that are on the five more difficult topics were further split into two subsets, one including the items on the cognitive domain App and Knw (to the left) and the other including the items on the cognitive domain Rsn (to the right; note in Table 1 that the average for Rsn is the largest among the three cognitive domains). And so on and so forth.

At each step during the construction of the tree, all possible splits on the basis of all possible predictor variables are considered. Possible splits are evaluated in terms of a *node-impurity measure* that measures the dissimilarity of the response variable among the observations belonging to each subset (e.g., Hastie et al., 2009, pp. 307–309). For regression problems, the variance of the response variable for a subset (also referred to as the residual sum of squares) is typically used as the node-impurity measure.

The best split is the one that produces the largest difference between the node-impurity measure of the original set and the sum of the node-impurity measures in the two potential subsets, or, equivalently, minimizes the sum of the node-impurity measures in the two potential subsets. For example, if the responses of the 100 individuals belonging to a node are integers between 1 and 100 in a regression problem, the best possible split[6] would divide the node into one node with individuals whose responses are 1–50 and another with individuals whose responses are 51–100. Thus, for example, the first split in Figure 1, which divides the items into one subset of easy items (to the left) and another subset of difficult items (to the right), makes sense. Typically, the predictor variable that is used in a split is the one that predicts the response the best and hence can separate the small responses from the large responses in the original set. For each split, the original set of observations is referred to as the *parent node* and the two subsets as the left and right child nodes. The node-splitting process continues until some *stopping criterion*, which is a rule that specifies when the process would stop, is reached. Examples of stopping criteria are stopping when the number of observations left in a node is smaller than a predetermined value and stopping when further splits would not reduce the node-impurity measure much (e.g., Strobl, 2013). Investigators often grow a large tree of, say, 50 nodes, and compare the cross-validation error of trees with, say, 10 nodes, 20 nodes, . . ., and 50 nodes, and choose the number of nodes that leads to the smallest cross-validation error (e.g., James et al., 2013, p. 308); this procedure is referred to as "pruning back." In the final tree, the nodes without a child are called *terminal nodes*. The tree in Figure 1 includes eight terminal nodes that divide the items into eight groups of varying average estimated difficulty.

Computation of predicted values from a regression tree. To obtain the predicted value of the response of an individual from the final fitted tree, one first determines the terminal node that corresponds to the predictors of the individual. Then the predicted value for the individual is obtained as the average of the responses of the individuals belonging to that terminal node (note that these averages are shown in the top line of the boxes in Figure 1). For example, Figure 1 shows that the terminal node that corresponds to the predictors of an item on topic NSW, cognitive domain Knw, and any reading demand is the first terminal node from the left. Then the predicted value of the item difficulty of such an item is obtained as −0.67.

Classification Trees: Construction and Computation of Predicted Values

Classification trees are constructed in a similar manner as regression trees. While building these trees, misclassification error, deviance, and Gini index,[7] all based on the frequency of the classes of the response variable for a subset, are used as the node-impurity measure (Hastie et al., 2009). If the responses of the 100 individuals belonging to a node are 50 0s and 50 1s in a classification problem, the best possible split would divide the node into one with the 50 0s and another with the 50 1s.

A CT for predicting the indicator of dropout from the available covariates for the dropout data set is provided in Figure 2. The figure looks very similar to Figure 1.

As with regression trees, of the two numbers in each box, the one at the bottom represents the number of observations corresponding to the node; the number at the top represents the most common value of the response variable among the individuals belonging to the node; it is also the corresponding predicted value if the node is a terminal node. For example, the predicted value for an examinee with RETENTION_PCT = 40 and AVG_DAYS_ABS = 20, from the third terminal node from the left (with the text 1 and 2% below it), is 1, that is, such an examinee is predicted to drop out from high school. The splits in Figure 2 make sense because they are based on the predictors RETENTION_PCT and AVG_DAYS_ABS that are the two most important predictors in LoR in Table 2.

Difference Between CARTs and Traditional Prediction Methods

While the prediction in MLR or LoR involves a linear function of the predictors (such as $\beta_0 + \beta_1\chi_1 + \beta_2\chi_2 + \cdots + \beta_p\chi_p$), the prediction in a CART involves partitioning of all observed combinations of values of the predictor variables in the training sample into regions $R_1, R_2, \ldots R_M$,[8] and then setting the predicted value as the discrete function

$$\text{Predicted value} = \begin{cases} c_1 \text{ if the combination of predictors is in Region } R_1, \\ c_2 \text{ if the combination of predictors is in Region } R_2, \\ \cdots \\ c_M if \text{ the combination of predictors is in Region } R_M. \end{cases}$$

For example, the box toward the top right of Figure 3 corresponds to the region where AVG_DAYS_ABS ≥ 12 and RETENTION_PCT ≥ 75 (this region corresponds to the second rightmost terminal node in Figure 2). The small gray 0s and 1s in the figure denote the actual values of the dropout indicators in the sample. For example, the gray 1 toward the top left indicates that a student with RETENTION_PCT = 20 and AVG_DAYS_ABS = 80 actually dropped out. A line of text

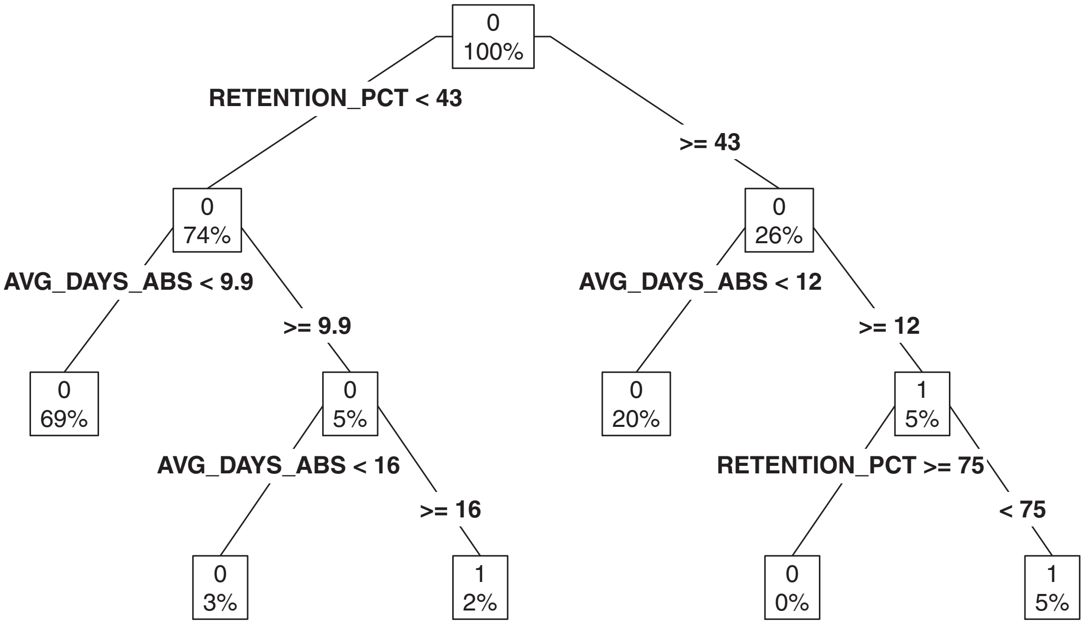

FIGURE 2. A classification tree for predicting dropout from available covariates.

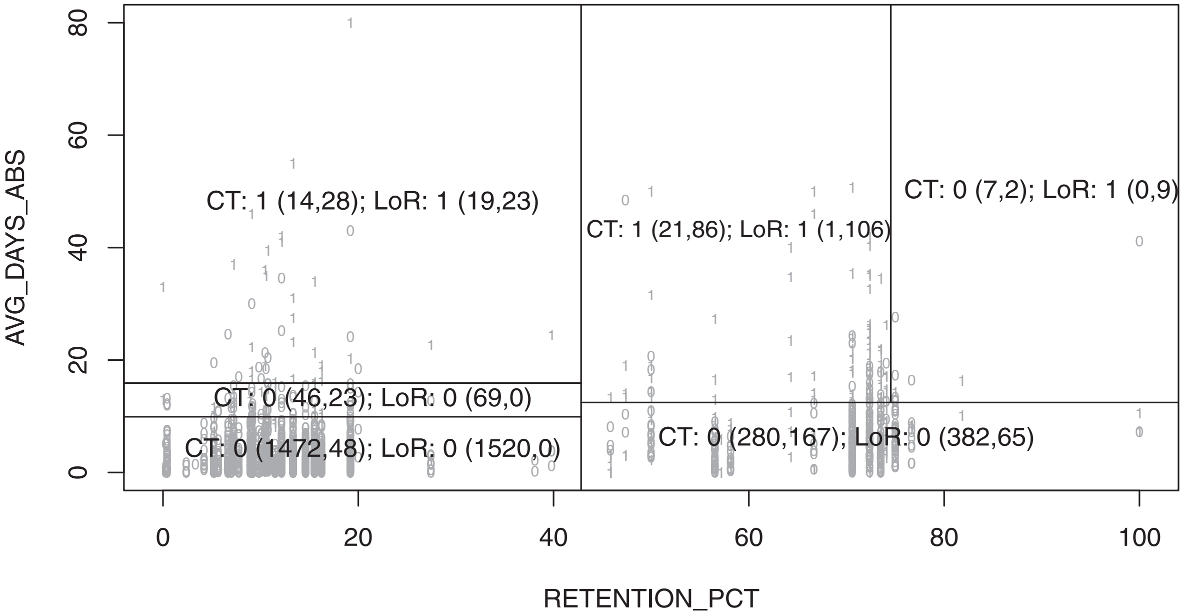

FIGURE 3. The regions corresponding to the classification tree for the dropout data. The regions R_1, R_2, . . . for the classification tree in Figure 2 are given.

in each box shows the corresponding predicted value from the CT, the observed number of 0s and 1s (under parenthesis), the most common predicted value for LoR, and the predicted number of 0s and 1s from LoR (under parenthesis). For example, "CT: 0 (7,2); LoR: 1 (0,9)" in the top right box indicates that for those in the region corresponding to the box, the predicted value from the tree, the actual number of 0s, and the actual number of 1s are 0, 7, and 2, respectively, and, the most common predicted value, and the predicted number of 0s and 1s from LoR are 1, 0, and 9, respectively. While prediction using a discrete function in a CART leads to some loss of information, the quality of prediction is decent in general and is better than that for MLR or LoR if the relationship between the predictors and the response is highly nonlinear (James et al., 2013, p. 314). For example, the predicted value from LoR is 1 ("drop out") for all the nine students corresponding to the top right box of Figure 3 because the LoR coefficient for both AVG_DAYS_ABS and RETENTION_PCT are positive and statistically significant in Table 2 and the box corresponds to large values of both of these predictors. However, the observed value is 1 for only two of these nine students—the relationship between the logit of the chance of a dropout and these two predictors does not seem to be linear in this region (if the relationship were linear, more students among these nine students would have dropped out, as predicted by LoR)—and the tree provides a better prediction for them.

Because of the way they *partition* the data *recursive*ly, the CARTs, as well as several other methods for supervised learning to be discussed later, are referred to as *recursive partitioning* methods. Regression trees have been applied in educational measurement by, for example, Sheehan and Mislevy (1994) for predicting the estimated difficulty of test items from the items' attributes.

While the CARTs have the advantage of requiring no assumptions about the distribution of the variables, a major problem with CARTs is their high variance or instability. A small change in the data often leads to a very different series of splits and hence to a tree that looks completely different from the original one. This instability may be overcome by constructing, instead of one tree, several trees and combining their predictions into a single prediction. The following two methods for supervised learning—random forests and boosting—implement this idea of using several trees (or, an *ensemble* of trees), and hence are referred to as *ensemble methods.* Figure 1 of Stijven, Minnebo, and Vladislavleva (2011) shows that while the prediction from one tree results in a discrete function, the ensemble methods can approximate continuous functions by combining many trees. In addition, the ensemble methods retain the trees' virtue of providing satisfactory prediction for nonlinear problems. An application of an ensemble method is similar to employing a committee of experts for prediction; a committee of experts with a diverse knowledge background is more likely than a single expert to make a correct prediction.

Random Forests

To apply random forests to CARTs, one draws B bootstrap samples from the training set where each bootstrap sample, whose size is the same as that of the training set, is drawn from the training set with replacement. One then constructs a tree using each bootstrap sample. To make the trees different from each other, each split in each tree belonging to a random forest is allowed to use only a random subset of the available p predictors. A fresh subset of predictors is chosen at each split. Typically, the number of predictors in these subsets is roughly equal to $\sqrt{p}$ for classification and $p/3$ for regression (e.g., Hastie et al., 2009, p. 592). Thus, in an application of random forests to the high school dropout data, each node would be allowed 3 (integer nearest to $\sqrt{12}$) predictor variables out of the 12. For regression trees, the predictions from the B individual trees are averaged to obtain the final prediction. For CTs, the final predicted class is obtained by taking a *majority vote* from the B trees, that is, by picking the most commonly occurring class among the predicted classes from the B trees (e.g., James et al., 2013, p. 317). Thus, for example, if the predicted class for a student is "no dropout" from 200 trees, but "dropout" from 300 trees, the final predicted class is "dropout."

Thus, the steps in constructing a random forest are:

- For $b = 1, 2, \ldots, B$, repeat:
  - Draw a bootstrap sample from the training set.
  - Construct a tree using the above bootstrap sample, by recursively repeating the following steps for each node of the tree, until a stopping criterion is reached:
    - *Randomly select m (out of p) predictor variables.
    - *Pick the best variable/split-point using the m predictor variables.
    - *Split the node into two child nodes using the above best variable/split-point.
- Compute the predicted value of the response for an observation as
  - average of the predicted values for the observation from the B trees for regression, and
  - the majority vote for the observation from the B trees for classification.

The stopping criterion is usually conservative so that each tree is large with several nodes and usually leads to a prediction with a high variance and low bias. Averaging of the predictions over the B trees leads to a small prediction variance.

Restricting the number of predictors at each split in random forests, which is referred to as *decorrelating*, makes the trees more different from each other than what they would be if all the predictors were allowed. Thus, an application of random forests is like employing a committee of experts whose knowledge backgrounds are diverse from each other because of first bootstrapping and then decorrelating—this diversification usually increases the chance of a correct overall prediction from the committee, especially for responses that are difficult to predict. In addition, the random selection of splitting variables in random forests allows each split to employ predictor variables that are less important than a stronger competitor but may reveal interaction effects that otherwise would have been missed (e.g., Strobl, 2013).

Figure 4 shows four trees from an application of random forests to the TIMSS data set. The first three trees employ the predictor "reading demand," which was not picked even once as a splitting variable in Figure 1 and hence is an example of a predictor that is "less important but may reveal interaction effects that otherwise would have been missed."

One can estimate the test error in random forests without performing cross-validation. The trees in random forests are constructed using nonparametric bootstrap samples, each of which includes some observations multiple times, of the training set. Roughly two-thirds of the observations of the training set appear in any bootstrap sample (e.g., James et al., 2013, p. 317), several appearing multiple times. For example, if the training set includes six observations, a bootstrap sample might include observations 1, 2, 2, 3, 3, 4—so it includes observations 2 and 3 twice each while it does not include observations 5 or 6; observations 5 and 6 are referred to as out-of-bag (OOB) observations for this bootstrap sample. After building a tree from this bootstrap sample, one can predict the responses of observations 5 and 6 using the tree and then compute prediction errors for them—these prediction errors are referred to as the OOB error. Because observations 5 and 6 were not used to build the tree, the OOB error is similar in concept to the test error. The aggregate of these OOB errors over all bootstrap samples, also referred to as the overall OOB error, is virtually equivalent to the cross-validation error when a large number of bootstrap samples are used (James et al., 2013, p. 318) and is a satisfactory estimate of the test error.

The number of trees B in random forests is predetermined or can be chosen using cross-validation or OOB error. *Bagging*, another of the methods for supervised learning, is a special case of random forests for $m = p$, that is, when each split in each tree is allowed to use all the predictors; bagging leads to trees that are less different from each other than in random forests. Random forests can be implemented using the R package *randomForest* (Liaw & Weiner, 2007), *party* (Hothorn, Hornik, Strobl, & Zeileis, 2014), and *caret* (Kuhn, 2008).

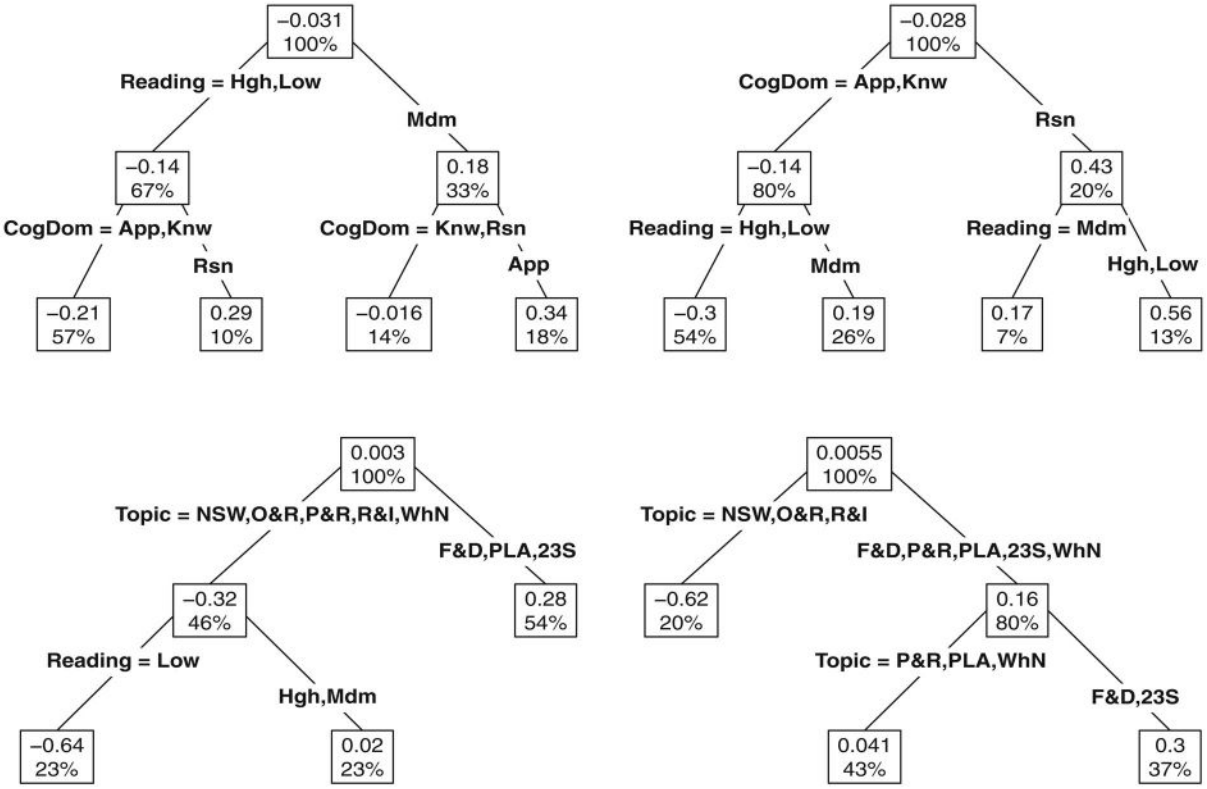

FIGURE 4. Illustration of the random forests procedure for the TIMSS data. The four panels show four different trees built from the data set.

Boosting

Boosting is similar to random forests in that it combines predictions from B trees. In the application of random forests, the investigator creates multiple copies of the training set by drawing bootstrap samples from it, fits a tree to each copy, and then combines the predictions from all the trees to obtain the final prediction. Each tree in these procedures is constructed separately from the other trees. However, in boosting, the trees are constructed sequentially and each tree is constructed using information from previously grown trees. No bootstrap samples are involved in boosting. Instead, each tree is constructed on a modified version of the original training set. There are several boosting algorithms. Among them, Adaboost (Freund & Schapire, 1997) and gradient boosting (Friedman, 2001) are arguably the most popular.

A boosting algorithm begins by building a “weak” initial model from the training set, where a “weak” model is one whose error rate is only slightly better than random guessing. Such a model can be obtained by, for example, splitting the data into two subsets based on one predictor. Figure 2 shows that a “weak” model for the dropout data can be obtained by splitting the examinees into two subsets based on RETENTION_PCT. Then, any observations in the training set that the model incorrectly classifies will have their importance boosted in the algorithm. This is done by assigning all observations a weight. All observations might start, for example, with a weight of 1 over the sample size. Weights are boosted through a formula so that the observations that are wrongly classified by a tree will have a larger weight while building the next tree.

A new tree is built with these boosted observations, which can be considered as problematic observations. The algorithm needs to take into account the weights of the observations in building the new tree. Consequently, the tree builder effectively tries harder to correctly classify the problematic observations. The process of building a tree and then boosting observations incorrectly classified is repeated until a newly generated tree performs no better than random guessing.

The steps in a boosting algorithm can be illustrated abstractly with a simple example. Suppose that the training set includes only five observations. Each observation will be assigned an initial weight of, say, .2. Suppose one builds a CT (Tree 1) that incorrectly classifies two observations (e.g., observations 4 and 5). One can calculate the sum of the weights of the misclassified observations as .4 (generally this is denoted as ε). This is a measure of the inaccuracy of this tree.

The ε is transformed into a measure used to update the weights and to provide a weight for the tree when it forms a part of the ensemble. This transformed value is α and is often something like $\log(\frac{1-\epsilon}{\epsilon})$. The updated weights of the misclassified observations are then calculated by multiplying the old weight by e^{α}. In our example, $\alpha = \log(\frac{1-0.4}{0.4}) = 0.405$, so that the updated weights for observations 4 and 5 become $.2e^{\alpha} = .3$. The weights for the correctly classified observations (observations 1, 2, and 3) remain unchanged at .2. The $\alpha = 0.405$ would later be assigned as the weight of Tree 1 when the tree would be included in the ensemble.

The tree builder algorithm is applied again under the condition that observations 4 and 5 are effectively multiplied to have larger weight or more representation in the algorithm. Thus, a different tree is likely to be built that is more likely to correctly classify these observations. Suppose that the new tree (Tree 2) incorrectly classifies observation 4 and correctly classifies the other four observations. The current weight of observation 4 is .3.

Thus, the new ε is .3. The new α is .85, which is larger than the α for Tree 1. It is again used to update the weights of the incorrectly classified observations, so that observation 4 gets an updated weight of $.3e^{\alpha} = .7$. The weights of the observations classified correctly by Tree 2 (observations 1, 2, 3, and 5) remain the same as before, at .2, .2, .2, and .3, respectively. So observation 4 has the largest weight now since it seems to be quite a problematic observation. The process continues for a large number of times or until the tree at a step has an error rate larger than 50%.

To obtain a prediction for a new observation from the ensemble of the trees, each tree is used to obtain a prediction for the observation. The final prediction is then calculated as the weighted average of these predictions, where the weight assigned to a tree is equal to the α associated with that tree. Note that the weight is larger for the trees that misclassify fewer observations.

Actual implementations of the boosting algorithm use variations of the simple approach (corresponding to the Adaboost.M1 algorithm; Freund & Schapire, 1997) that is illustrated above. Variations are found in the formulas for updating the weights and for weighting the individual models. However, the overall concept remains the same.

Thus, the application of boosting is like employing a committee of experts whose knowledge backgrounds are less different than in random forests. However, while the experts do not learn from each other and receive equal importance in random forests, boosting allows Expert 2 to learn from the mistakes of Expert 1 (or, to try to predict harder the observations incorrectly predicted by Expert 1), Expert 3 to learn from the mistakes of Expert 2, and so on, and, in the end, gives more weight to the predictions of the experts who made fewer mistakes.

The number of trees (B) in boosting is predetermined or chosen from a K-fold cross-validation. Often, trees with only two terminal nodes (or, a "stump") work well (e.g., James et al., 2013, p. 323). Boosting can be implemented using the R packages *ada* (Culp, Johnson, & Michailidis, 2006), *caret* (Kuhn, 2008) and *gbm* (Ridgeway, 2014).

Further Details on the Methods for Supervised Learning

Model Assumptions

The traditional prediction methods such as MLR depend on assumptions such as linearity, homogeneity of variance, and normality of the observations. The methods for supervised learning, which do not depend on any probability model and are algorithmic in nature, are expected to perform better than traditional prediction methods for data sets for which these assumptions do not hold (e.g., James et al., 2013; Strobl, 2013). However, as discussed later, the methods for supervised learning have some limitations. Note that when the traditional methods are inappropriate, one can use nonlinear regression, which also have some limitations such as poor convergence and requirement of special computer programs (e.g., Draper & Smith, 1998, p. 505).

Computation and Additional Software Packages

All methods for supervised learning are computation-intensive and require specific computer packages for implementation. However, there are several publicly available and free software packages for implementing these methods. Further, the methods for supervised learning can be applied to large data sets in a timely manner. In addition to the above-mentioned R packages, it is also possible to implement CARTS, random forest, and boosting using *Rattle* (e.g., Williams, 2011), which is an R package but provides a simple and intuitive interface that allows users to load and explore the data and build and evaluate several methods for supervised learning. Freely available data mining tools are available from the WEKA data mining software (Hall et al., 2009) and in the scikit-learn library (e.g., Pedregosa et al., 2011) for the Python programming language.

Note that most of the methods for supervised learning involve random sampling and will produce slightly different results if the same model is fitted to the same data set on two computation runs. To obtain exactly the same result on different runs, one can set the random seed, which determines the internal random number generation of the computer, to a fixed value (e.g., Strobl, 2013).

Theoretical Results in Favor of the Methods for Supervised Learning

Though most of the support in favor of the methods for supervised learning has been obtained by examining the test error of the methods for real data sets, several theoretical results are available on the methods for supervised learning. These results prove, for example, the consistency and other statistical properties of boosting (Buhlmann & Yu, 2003; Friedman, 2001; Friedman, Hastie, & Tibshirani, 2000) and consistency of prediction from random forests and other ensemble methods (Breiman, 2001), and explain why the decorrelation involved in random forests usually leads to better prediction (Breiman, 2001). These research articles have contributed to a deeper understanding of the statistical background behind several methods for supervised learning and facilitated their increasing popularity.

The Predictor Variables and Variable Importance Measures

In traditional prediction methods such as MLR, the relevance or importance of a predictor variable can be measured by, for example, the corresponding estimated regression coefficient and its SE. In random forests and boosting, regression coefficients or SEs do not exist, but *variable importance measures* can be computed to assess the relevance of each predictor variable over all trees of the ensemble. One measure of importance of a predictor is the total reduction in the aforementioned node-impurity measure due to the splits involving the predictor, averaged over all the trees (e.g., Friedman, 2001; James et al., 2013, p. 319). It is expected that an important predictor would be used in many trees in many splits to cause substantial reductions in the node-impurity measure. For example, the predictor "topic" is used to split the nodes several times over two trees in Figure 4 to create child nodes that have substantially different averages; naturally, the variable importance of "topic" is the largest for random forests for the data set. Typically, investigators use as a variable importance measure the average reduction in variance for regression problems and the average reduction in the "Gini index" for classification problems, where either of these is computed for each predictor variable over all the nodes in all the trees (e.g., James et al., 2013, p. 312). The variable importance measure will be larger for more important predictors. In

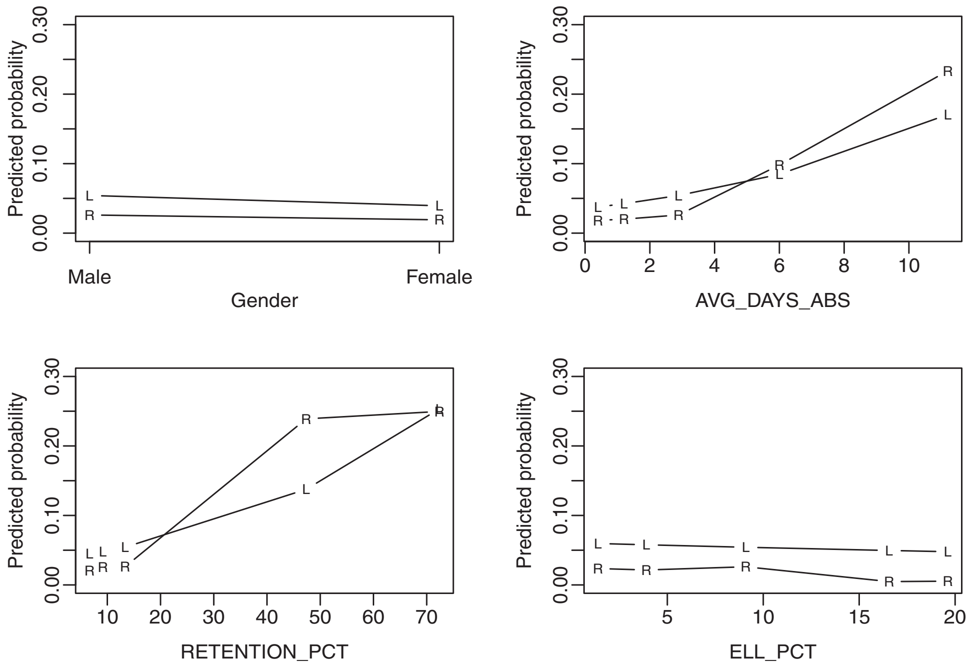

FIGURE 5. A sensitivity plot for the dropout data showing how the predicted probabilities of dropout change as one predictor changes while all the other predictors are held fixed.

random forests, the OOB observations can be used to construct another variable importance measure that is referred to as the *permutation importance* (e.g., Strobl, 2013).

Which Among the Methods for Supervised Learning Should Be Used in an Application?

No one among the methods for supervised learning dominates all others over all possible data sets (James et al., 2013, p. 29)—so it is impossible to recommend any one of them for all prediction problems in educational measurements. However, random forests and boosting have performed satisfactorily in several prediction problems in several scientific fields. Random forests outperformed several other methods for supervised learning such as boosting and bagging in the context of classification of cancer prevalence based on mass spectrometry in Wu et al. (2003). Boosting and random forests outperformed LoR and several other methods for supervised learning in Caruana, Karampatziakis, and Yessenalina (2008) who considered 11 data sets from various fields such as medicine, chemistry, ornithology, behavior, and image processing. Svetnik, Liaw, Tong, and Wang (2004) found random forests to outperform several other methods for supervised learning for six data sets for Quantitative Structure–Activity Relationship modeling for pharmaceutical molecules. Fernandez-Delgado et al. (2014) found random forests to be the best classifier in a comparison of more than 100 methods for supervised learning using 121 data sets from various fields; the average accuracy of random forests was 82% while that of LoR was 78%.

In contrast, Konig et al. (2008) found that boosting and random forests fared quite similarly while LoR performed significantly better in predicting binary patient outcomes.

Other Data Mining Methods

The methods for supervised learning that were not covered here include the nearest neighbor methods, kernel-smoothing methods and local regression, generalized additive models, and multivariate adaptive regression splines. Neural networks and support vector machines were considered, but they performed worse than traditional prediction methods. Bagging was considered, but performed slightly worse than random forests in each example. "Stacking" (e.g., Hastie et al., 2009, p. 290), which involves a combination of predictions from multiple methods for supervised learning to improve performance, was not considered here. The unsupervised-learning methods, in which the investigator has the values of a set of variables for a sample of individuals and tries to learn about the joint density of the variables (and does not attempt to predict the values of any variable), were not covered in this module. Examples of such methods are association rule learning or association rules, clustering methods, principal components analysis, principal curves, self-organizing maps, multidimensional scaling, and mixture modeling.

Limitations of the Methods for Supervised Learning

Though the methods for supervised learning are of increasing interest, they have several limitations. The first limitation is that even with increased computational power, the methods for supervised learning are more time-consuming than traditional prediction methods and could take hours for large data sets compared to seconds for traditional methods.

Second, unlike traditional prediction methods, the methods for supervised learning may account for the true functional form of the relationship between the response and the predictor variables, however complex the relationship may be. However, the downside is that the relationship is usually impossible to interpret by humans. Thus, the methods for supervised learning are often like *black boxes* so that the investigator feeds the data and receives predictions for the test set, but does not have any idea about the exact form of relationship between the response and predictor variables.

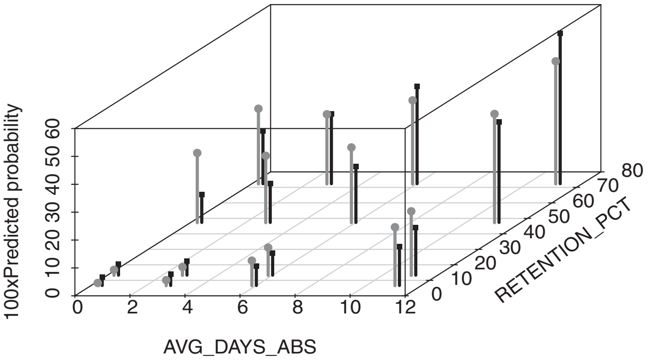

FIGURE 6. A sensitivity plot showing how two predictors influence the dropout rate.

In fact, James et al. (2013, p. 319) noted that the methods for supervised learning improve prediction accuracy at the expense of interpretability. However, there have been some recent attempts such as Cortez and Embrechts (2013) to increase interpretability of the methods for supervised learning using "sensitivity analysis"—two examples of such analysis are provided in Figures 5 and 6.

Third, as mentioned earlier, some theoretical results are available for the methods for supervised learning, but the number of such results pales in comparison to that for traditional prediction methods. For example, in MLR, theoretical results are available on the SE and the distribution of the estimated regression coefficients and the predicted values, model fit, and sample size requirements. However, such results are lacking for the methods for supervised learning because they are not based on probability and mathematical statistics and are algorithmic in the sense that they rely on computer programs to determine the functional form in an exhaustive search.

Finally, with hundreds of methods for supervised learning available in statistics and computer science literature (and more evolving from current research) and a research history of about 10–15 years, it is a daunting task for even an experienced investigator to apply these methods and find the appropriate methods for supervised learning and the appropriate software packages for the problem at hand.

Applications

The above-mentioned methods for supervised learning were applied to the TIMSS data, the high school dropout data, and the electronic essay scoring data. For each data set, the following steps were replicated 1,000 times:

- The data set was randomly split into a training set of about two-thirds of the total number of observations and a test set with the remaining one-third observations.
- The prediction models for the methods for supervised learning were built from the training set.
- The prediction models were used to predict the response of each observation in the test set.
- Several test-set accuracy/error measures were computed from the predicted and actual values of the response variable in the test set.[9]

The average of each test set accuracy/error measure was then computed from the 1,000 replications. The random forests and boosting procedures involved 500 trees, which is the default option in the R software packages used to implement them. Increasing the number of trees had a negligible effect on the average test-set accuracy.

Predicting Item Parameters From Item Attributes

Columns 3 and 4 of Table 3 provide the average (over 1,000 replications) of RMSE and correlation coefficient computed from the test set from the prediction of estimated item difficulty from the item attributes. Columns 5 and 6 of the table show similar results for prediction of estimated item slope from the item attributes. Smaller values of RMSE and larger values of the correlation coefficient indicate better performance of the corresponding method.

All the computations were performed using R software packages that are listed in the table.[10] Appendix A includes R code for some of these computations. Table 3 includes the results for the traditional method of stepwise MLR (or, equivalently, analysis of variance) where main effects (such as "topic" and "cognitive domain") and first-order interactions (such as "topic" and "cognitive domain") were allowed.[11] The table shows that random forests performed the best followed by boosting—they had the smallest RMSE and the largest correlation coefficient. A single (regression) tree also performed slightly better than MLR. However, the improvement of the methods for supervised learning over MLR was modest and most likely of no practical consequence (predicted item parameters are typically used to construct test forms; slightly better prediction would not lead to much practical gain).

The variable importance measure is largest for the predictor "topic," followed by "cognitive domain," which is in agreement with the aforementioned results for the data set from MLR.

Predicting High School Dropout

Table 4 provides the R software packages used and the average over the 1,000 replications of the following test-test accuracy measures for the dropout data:

- CCR and Cohen's kappa.
- "Sensitivity" (or "true positive rate"), which is the proportion of actual dropouts who are correctly predicted as dropouts.
- "Specificity" (or "true negative rate"), which is the proportion of actual nondropouts who are correctly predicted as nondropouts.

For all of these measures, larger values indicate better performance. The results for the traditional methods of LoR and multilevel LoR (Subedi & Howard, 2013) with only the main effects of the predictors[12] are also included in the table.

The table shows that boosting and random forests slightly improved over the traditional approaches of LoR and multilevel LoR. Random forests performed the best. A single CT also performed slightly better than LoR. According to National Center of Education Statistics, an estimated 14.7 million of students attended high school (i.e., Grades 9–12) in Fall 2014 (nces.ed.gov/fastfacts/display.asp?id = 372). Therefore, though the extent of improvement in CCR (or any other accuracy measure) from LoR to, for example, random forests is only .03 (also, this improvement is statistically significant when the SE from the 1,000 replications is taken into account), this slight improvement could translate to more accurate prediction of dropout status (followed by appropriate intervention) for hundreds of thousands of high school students. Thus, the

Table 3. Average of the Accuracy Measures for the TIMSS Data

		Item Difficulty		Item Slope	
Method	R Package	RMSE	Correlation	RMSE	Correlation
MLR	lm	.73	.30	.28	.42
Single regression tree	rpart	.72	.31	.27	.44
Random forests	randomForest	.69	.35	.26	.45
Gradient boosting	gbm	.70	.34	.27	.43

Note. The results from analysis of variance (using the R function "aov") are identical to the above results from MLR.

Table 4. Average of the Accuracy Measures for the High School Dropout Data

Method	R Package	CCR	Sensitivity	Specificity	Kappa
LoR	glm	.85	.88	.58	.41
Multilevel LoR	lme4	.86	.92	.57	.49
Single classification tree	rpart	.86	.91	.58	.46
Random forests	randomForest	.88	.91	.65	.51
Gradient boosting	gbm	.87	.90	.66	.45

methods for supervised learning promise to lead to a substantial practical benefit compared to traditional methods in the context of predicting dropout from high schools.

The last column of Table 2 (with the column heading "Gini") provides the average of the mean decrease in the Gini index from the fitting of the random forests to the 100 training sets. These values have been expressed as a percentage of the maximum so that the value is 100 for "average days absent" (that seems to be the most important predictor) and 45 for "percent retained of school" (the second most important predictor). Note that these two variables are the two most significant predictors in LoR as well (with large z-values and small p-values). The z-values from LoR and the average decrease in the Gini index mostly agree for other continuous predictors as well. Interestingly, the two examinee-level FCAT scores are the third and fourth most important variables and the one school-level FCAT score is the fifth most important variable according to the Gini index; thus, the FCAT scores, as in LoR, are not the most important predictors in random forests either. This could be due to the fact that this data set only included at-risk students whose FCAT scores are in Levels 1 and 2 (our of a possible 1–5), that is, restricted in range.

Figure 5 shows the results of a sensitivity analysis (e.g., Cortez & Embrechts, 2013) to increase interpretability of the methods for supervised learning. Each panel in the figure shows how the predicted probabilities of dropout (i.e., also the expected value of the response variable "dropout indicator"[13]) from LoR and random forests change when all the predictors are held at their median values[14] except one that varies between its possible values if it is discrete (like "gender") or varies between its 10th, 25th, 50th, 75th, and 90th percentiles if it is continuous (like AVG_DAYS_ABS).[15] In the top left panel, the line joining the two "L"s represents the predicted probabilities from LoR for two levels of the predictor "gender" ("male" and "female") and the line joining the two "R"s represents the corresponding predicted probabilities from random forests when all the other predictors are fixed at their median values. The predicted probability for females is slightly smaller than that of males for both methods, which agrees with the negative LoR coefficient in Table 2 for "Gender." In the top right panel, the line joining the five "L"s represents the predicted probabilities from LoR and the line joining the five "R"s denotes the predicted probabilities from random forests when AVG_DAYS_ABS varies over its 10th, 25th, 50th, 75th, and 90th percentiles with all the other predictors fixed at their median values. The bottom two panels show similar plots for RETENTION_PCT and ELL_PCT. The range of the vertical scale is the same in the five panels. A regression coefficient in MLR is interpreted as the predicted/expected change in the response when the corresponding predictor is increased by 1 keeping the other predictors fixed (e.g., Draper & Smith, 1998, p. 42). The methods for supervised learning do not involve any coefficients such as regression coefficients.

However, the examination of how the expected response changes as the result of a change in a predictor when all the other predictors are fixed at their average values, as in Figure 5, provides some insight on how the response is influenced by the predictor in applications of the methods for supervised learning.

The pattern of the predicted probabilities is similar between LoR and random forests. For example, the predicted probability steadily increases with an increase in either of AVG_DAYS_ABS and RETENTION_PCT, but stays mostly flat with an increase in ELL_PCT both for LoR and random forests (this also agrees with the p-value and Gini index for these three predictors). The figure also points to a difference between LoR and random forests; whereas the predicted probabilities smoothly increase for LoR, they often increase sharply for random forests, for example, between the 50th and 75th percentiles (the third and fourth "R" from the left) of AVG_DAYS_ABS and RETENTION_PCT. The flatness of the predicted value from random forests between the 75th and 90th percentiles in the bottom left panel is most likely related to the phenomenon of only two dropouts out of a possible nine in the top right box of Figure 3. Such flexibility of the random forests most likely contributes to their improved prediction power over traditional prediction methods that are restricted to involve linear or logistic relationships between the response and the predictors.

Figure 6 shows a three-dimensional plot showing how the predicted probability of dropout changes as all the predictors are held at their median values except AVG_DAYS_ABS and RETENTION_PCT (the two most important predictors for these data) that vary between their 10th, 50th, 75th, and 90th percentiles. The values of AVG_DAYS_ABS are shown

Table 5. Average of the Accuracy Measures for the Electronic Essay Scoring Data

		Large Training Set		Small Training Set	
Method	R Package	CCR	QWK	CCR	QWK
MLR	lm	.76	.74	.65	.63
Cumulative LoR	polr	.79	.79	–	–
Single Classification tree	rpart	.77	.78	.74	.75
Random forests	randomForest	.82	.85	.80	.82
Gradient boosting	gbm	.81	.84	.79	.80

along the X-axis, those of RETENTION_PCT along the Y-axis, and 100 times the predicted probabilities from LoR (black vertical bars with a solid black square at the top) and random forests (gray vertical bars with a solid gray circle at the top) are shown along the Z-axis.[16] For example, the two vertical bars at the top right corner show that when both of these predictors are equal to their 90th percentiles (and the other predictors are equal to their medians), the predicted probabilities are .54 and .44, respectively, for LoR and random forests. The figure shows that while predicted values from both methods increase as either of the two predictors increases, there are some differences. While the probabilities for random forests are larger than those for LoR toward the top left corner, the opposite phenomenon is observed at the top right corner (where both the predictors are large), which is related to the phenomenon of LoR overestimating dropout for the examinees in the top right box of Figure 3. In addition, as in Figure 5, the predicted probabilities for LoR increase more smoothly than those for random forests.

Electronic Essay Scoring

Columns 3 and 4 of Table 5 provide the R software packages used and the average (over 1,000 replications) CCR and the QWK (Cohen, 1968) for the test set for several methods for supervised learning. The results for MLR and cumulative LoR[17] are also included in the table.

The table shows that boosting and random forests performed the best and slightly outperformed the traditional prediction methods, as indicated by the improvement in CCR from .76 or .79 (MLR or LoR) to .81 or .82. With an increasing popularity of performance tasks (e.g., Darling-Hammond & Adamson, 2010, p. 1) that are almost always rated, the above improvement in prediction by the methods for supervised learning, which would lead to more fair scores, promises to lead to a substantial practical benefit.

Researchers such as Strobl et al. (2009, p. 324) stressed the advantages of the methods for supervised learning over traditional classification and regression methods for massive data sets and, especially, high-dimensional problems or "$p >> N$" problems in which the number of predictors (p) is large compared to the number of observations (N). Strobl et al. (2009) applied random forests to predict the occurrence of bipolar disorder from a data set with 61 individuals and 102 predictors; application of traditional methods such as LoR is problematic for such a data set. To examine what to expect for high-dimensional problems, the above-mentioned computational steps for the electronic essay scoring data set were performed with much smaller training sets. The average test-set accuracy over 1,000 replications for several methods is provided in Columns 5 and 6 of Table 5 when the training set consisted of 100 observations (as opposed to 620 earlier), the test set consisted of the rest of the sample, and all the 49 available essay features were used as predictors.

The LoR could not be performed in this case because of the lack of enough observations for each possible value of the response variable. The improvement provided by the methods for supervised learning over MLR[18] for small training sets is much more than that for larger training sets. In practice, a few hundred to a couple of thousands of essays are typically used for constructing the prediction model in electronic essay scoring (e.g., Ramineni & Williamson, 2013, p. 30). Therefore, Columns 5 and 6 of Table 5 indicate that the methods for supervised learning may allow the construction of prediction models in electronic essay scoring from training sets much smaller than that currently used, which may lead to a substantial saving of resources.

Conclusions and Recommendations

This module provides a nontechnical review of several methods for supervised learning. Three real data sets from educational measurement were then used to illustrate the application of several methods for supervised learning using freely available R software packages. Several of the methods for supervised learning slightly outperformed the traditional prediction methods in the examples. The random forests performed the best overall and outperformed the traditional prediction methods in all the examples, which is in agreement with the findings of Fernandez-Delgado et al. (2014) and Wu et al. (2003). Boosting also outperformed the traditional prediction methods in the examples. The computation times of the methods for supervised learning were very short and of the order of a couple of seconds for these data sets.

Note that the extent of improvement that the methods for supervised learning provide over the traditional prediction methods is modest in the examples in this module. This result and similar results from other fields (e.g., Fernandez-Delgado et al., 2014) indicate that measurement practitioners should not expect huge improvements from the methods for supervised learning. In addition, the methods for supervised learning cannot provide excellent prediction from data that are of bad quality (e.g., Nisbet, Elder, & Miner, 2009, pp. 734–735). Therefore, even when tools for implementing the methods for supervised learning are available, the investigators should try their best to collect data of good quality. Also, the above results do not guarantee that random forests (or boosting) will be the best for all data sets in educational measurement. In a real application, it would be prudent to apply several methods for supervised learning and choose the one that minimizes cross-validation error.

An important practical question is "When should one consider applying the methods for supervised learning in educational measurement?" Given the subtleties and idiosyncratic nature of the classification and regression problems in educational measurement, it is difficult to provide a generalizable and precise answer to this question. However, based on applications of several methods for supervised learning to the data sets in this module and to data in other fields, one can conclude that the methods for supervised learning are worth considering when at least one or more of the following is true in a prediction problem:

- The assumptions of MLR and/or LoR may not hold. For example, a CT provided better prediction compared to LoR in the top right box of Figure 3 where the assumptions of LoR do not seem to hold. Other examples of violation of the assumptions of MLR and/or LoR in educational measurement are nonlinearity of the regression of GRE Verbal scores on the TOEFL scores (e.g., Stricker, 2002) and the lack of linearity in the context of growth measurement (e.g., Betebenner, 2009).
- The problem is high-dimensional. The better performance of the methods for supervised learning compared to MLR for training set size of 100 in Table 5 provides support to this. Other examples of high-dimensional problems in educational measurement are identification of at-risk students (e.g., West, 2012), personalized learning including game-based learning (e.g., Lin, Yeh, Hung, & Chang, 2013), and massive open online courses (e.g., Perna et al., 2014; Tucker, Pursel, & Divinsky, 2014), all of which could lead to a large number of variables for each student. With the advent of low-cost computing and network infrastructure, the number of such applications promises to increase. For example, U.S. Department of Education's February 2013 report "Promoting Grit, Tenacity, and Perseverance: Critical Factors for Success in the 21st Century" mentions (p. 32) that "With the prevalence of new digital learning resources and learning technologies, new forms of measurement are emerging, making it possible to go beyond conventional approaches. For example, data mining techniques can track students' trajectories of persistence and learning over time, thereby providing actionable feedback to students and teachers." Similar statements were made by U.S. Department of Education, Office of Educational Technology (2012).
- The performance of traditional prediction methods is inadequate and the investigator has the time and resources to explore alternative methods.

The methods for supervised learning are not without limitations, however; some of these were discussed earlier. In addition, while the sensitivity analysis in Figures 5 and 6 provided some insight of how the methods for supervised learning work and how they are different in nature from LoR, the extent of that insight is much smaller than that obtained from traditional prediction methods. Such a lack of insight may make domain experts nervous about the use of the methods for supervised learning. Further, while Table 2 shows a variable importance measure (based on the Gini index) from random forests, the p-values from LoR provide more information in the form of a formal statistical test of the importance of the covariates. Finally, while the run-time of the methods for supervised learning for the data sets here was negligible, it could be much more for huge data sets, for example, those from state tests.

Several related topics can be examined further. First, one could consider the data mining methods such as generalized additive models and multivariate adaptive regression splines that were not included here. Second, one could compare the methods for supervised learning for more data sets from various areas of educational measurement. Of special interest would be high-dimensional or "$p >> N$" problems. Third, more research can be performed to find a more precise answer to the above-mentioned question "When should one consider applying the methods for supervised learning in educational measurement?" On a related note, researchers could try to find the type of measurement problems for which a specific method (say, random forests) is expected to perform better than traditional methods. Fourth, it would be of interest to examine whether the quality of prediction in the methods for supervised learning is uniform over subgroups of individuals or whether the quality is better for certain subgroups; this question is somewhat related to the issue of differential validity. Fifth, while this module included some analysis on the interpretability of the methods for supervised learning, more such research would be useful to measurement practitioners. Finally, more collaborative research between the two communities, educational data mining and educational measurement, may be beneficial to both these communities and to numerous examinees or test-score users.

Acknowledgments

The author would like to thank the editor Derek Briggs, the ITEMS editor W. Holmes Finch, and the reviewers for their helpful comments. The author would also like to thank Vincent Kieftenbeld and Bidya Raj Subedi for their help with two of the data sets used in this module.

Appendix A: Annotated R Code for the TIMSS Data

```
>library(rpart)
>library(randomForest)
>library(gbm)
>D = read.csv('Data.csv',header = TRUE) #Read the input
  file
>Y = D$slope #Change to D$diff to predict item difficulties
>Topic = as.factor(D$topic)
>CogDom = as.factor(D$cog_domain)
>Read = as.factor(D$read_demand)
>Data = data.frame(Y,Topic,CogDom,Read)
>s = sample(1:nrow(Data),110)#Assign a subset of the data
  as the Training set
>Train = Data[s,]
>Test = Data[-s,]#Assign the other subset of the data as the
  Test set
# LINEAR REGRESSION
>reg = lm(Y~.,data = Train)
#For stepwise regression, use "step = stepAIC(reg,direction
  = "both",trace = FALSE)"
>predReg = predict(reg,newdata = Test)
>sqrt(mean((Test$Y-predReg)**2))#Compute RMSE from
  the test set for linear regression
# REGRESSION TREE
>CT = rpart(Y~.,data = Train)
>predTree = predict(CT,newdata = Test)
>sqrt(mean((Test$Y-predTree)**2))
# RANDOM FOREST
>RF = randomForest(Y~.,data = Train,ntree = 500)
>predRF = predict(RF,newdata = Test)
```

```
>sqrt(mean((Test$Y-predRF)**2))
#BOOSTING
>gbm1 = gbm(Y~.,data = Train,distribution = "gaussian",n.trees = 500,shrinkage = .05)
>predBoost ← predict(gbm1,newdata = Test,type = "response",n.trees = 200)
>sqrt(mean((Test$Y-predBoost)**2))
```

Appendix B: Self-Test

1. Apply the R code provided in Appendix A to the data provided in the online repository. Try slight variations such as different size of the training set, inclusion of interactions among predictors, application of stepwise regression, different number of trees, and so on.
2. Apply the R code provided in Appendix A (or code you found from another source) to apply boosting, random forests, and linear/logistic regression to a data set (with a response variable and one or more predictor variables) that you have.
3. Which of the following methods is not one for supervised learning?
   a Hierarchical clustering.
   b Principal component analysis.
   c Principal component regression.
   d Linear regression.
   e Neural networks.
4. Do you know of any applications of methods for supervised learning in the annual meeting of the NCME in 2015?
5. What two strategies are used to introduce randomness in random forests?

Answers to the Questions on the Self-Test

1. The root mean squared errors (RMSE) should not be exactly equal to those in Table 3 that were obtained from several iterations, but should be close to those. For example, the RMSE should be between .6 and .8 for the methods for supervised learning for prediction of item difficulty and between .2 and .4 for prediction of item slope.
2. If your data set has only a few predictor variables, then the methods for supervised learning are not expected to predict/classify any better than linear/logistic regression.
3. Hierarchical clustering and principal component analysis. All the other methods predict a response variable and hence a method for supervised learning.
4. There were several applications. For example, Jeffrey Steedle and Steve Ferrara applied regression tree and random forests to predict item parameters, Scott Wood and Sue Lottridge applied random forests to predict essay scores, and Jing Chen, James Fife, and Mo Zhang applied random forests, k-nearest neighbors, and support vector machines to predict essay scores. Thus, it seems that the methods for supervised learning are becoming popular in educational measurement.
5. (i) Bootstrap and (ii) the choice of a random subset of predictors at each split.

Notes

[1] Romero and Ventura are two of the four editors of the popular *Handbook of Educational Data Mining*.

[2] Note that because reading demand has three levels, only two indicator variables should be used in the MLR to avoid multicollinearity (e.g., Shieh & Fouladi, 2003).

[3] Or, equivalently, a three-factor analysis of variance model with each item attribute as a factor.

[4] As is typical with LoR models, a predicted value was set to 1 (dropout) when the corresponding estimated probability of a dropout from the fitted model was larger than .5 and set to 0 otherwise.

[5] This is because prediction models for the methods for supervised learning are built on the training set and the models try hard to fit all observations of the training set, some of which may be different from each observation in the test set.

[6] Note that this split would occur only if there is a predictor that predicts the response perfectly.

[7] The Gini index is a measure of the total variance across the classes. If p^{mk} denotes the proportion of observations in a node that are from the kth class, then the index for the node is defined as $\Sigma_k\, p^{mk}\ (1 - p^{mk})$.

[8] Where, for example, R_1 could be $\chi_1 < 4,\ \chi_2 < 1,\ \chi_3 = 0, \ldots,\ \chi_p > 2$.

[9] It is possible in each replication to (1) fit, for example, several random forests (with different number of trees) using K-fold cross-validation on the training set, and pick the "best" random forest as the one that minimizes the K-fold cross-validation error, and (2) compute test-set accuracy measures of this "best" random forest. This strategy, some limited analyses shows, leads to a slightly better prediction for the methods for supervised learning. However, this was not performed to simplify the analyses.

[10] For this example and the other examples, the R package "party" (Hothorn et al., 2014) was used to implement the CARTs and random forests, but the quality of prediction was slightly worse—so results for the randomForest package are shown in this module.

[11] Other variations of the MLR, such as allowing the second-order interaction among the attributes, not allowing any interactions, and not using stepwise regression, did not improve its performance.

[12] Inclusion of the first-order interaction or the stepwise option did not improve the quality of the prediction.

[13] This is because the expectation of a 0–1 variable is the probability of its being equal to 1.

[14] Computed from the whole data set.

[15] The predicted probabilities for random forests were obtained from the "predict" function (with type = "prob") on a random forest object produced by the R package "randomForest." The input to the "predict" function included the values of the predictor where predictions were intended.

[16] Note that the gray and black vertical bars should have been drawn on top of each other; however, they are slightly separated for the convenience of viewing.

[17] Both with only the main effects of the predictors and the stepwise option. Inclusion of the first-order interaction was not possible due to too many predictors; the quality of the prediction was slightly worse without the stepwise option.

[18] The average CCR of stepwise MLR was only .02 more than that of MLR.

References

Abu-Oda, G. S., & El-Halees, A. M. (2015). Data mining in higher education: University student dropout case study. *International Journal of Data Mining and Knowledge Management Process*, *5*(1), 15–27.

Antunes, C. (2011). Anticipating students' failures as soon as possible. In C. Romero, S. Ventura, M. Pechenizky, & R. S. J. d. Baker (Eds.), *Handbook of educational data mining* (pp. 353–364). Boca Raton, FL: Chapman and Hall.

Baker, R. S. J. d. (2010). Data mining for education. In B. McGaw, P. Peterson, & E. Baker (Eds.), *International encyclopedia of education* (3rd ed.) (pp. 112–118). Oxford, UK: Elsevier.

Betebenner, D. (2009). Norm- and criterion-referenced student growth. *Educational Measurement: Issues and Practice*, *28*(4), 42–51.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth.

Buhlmann, P., & Yu, B. (2003). Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association*, *98*, 324–339.

Burrus, J., & Roberts, R. D. (2012). *Dropping out of high school: Prevalence, risk factors, and remediation strategies* (Educational Testing Service R & D Connections No. 18). Princeton, NJ: Educational Testing Service.

Caruana, R., Karampatziakis, N., & Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In W. Cohen, A. McCallum, & S. Roweis (Eds.), *Proceedings of the 25th Annual International Conference on Machine Learning (ICML)* (pp. 96–103). New York, NY: ACM Press.

Cohen, J. (1968). Weighed kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*, 213–220.

Cortez, P., & Embrechts, M. J. (2013). Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, *225*, 1–17.

Culp, M., Johnson, K., & Michailidis, G. (2006). ada: an R package for stochastic boosting. *Journal of Statistical Software*, *17*(2), 1–27.

Darling-Hammond, L., & Adamson, F. (2010). *Beyond basic skills: The role of performance assessment in achieving 21st century standards of learning*. Technical Report. Stanford, CA: Stanford University, Stanford Center for Opportunity Policy in Education.

Draper, N. R., & Smith, H. (1998). *Applied regression analysis* (3rd ed.). New York, NY: John Wiley.

Fernandez-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, *15*, 3133–3181.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*, 119–139.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*, 1189–1232.

Friedman, J. H., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, *28*, 337–407.

Gierl, M. J. (2007). Making diagnostic inferences about cognitive attributes using the rule space model and attribute hierarchy method. *Journal of Educational Measurement*, *27*, 325–340.

Gorin, J. S., & Embretson, S. E. (2006). Item difficulty modeling of paragraph comprehension items. *Applied Psychological Measurement*, *30*, 394–411.

Haberman, S. J., & Sinharay, S. (2010). The application of the cumulative logistic regression model to automated essay scoring. *Journal of Educational and Behavioral Statistics*, *35*, 586–602.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, *11*(1), 10–18.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. New York, NY: Springer.

He, Q., & Veldkamp, B. P. (2012). Classifying unstructured textual data using the Product Score Model: An alternative text mining algorithm. In T. J. H. M. Eggen & B. P. Veldkamp (Eds.), *Psychometrics in practice at RCEC* (pp. 47–62). Enschede, The Netherlands: RCEC.

Hothorn, T., Hornik, K., Strobl, C., & Zeileis, A. (2014). *Party: A laboratory for recursive partitioning*. Retrieved June 10, 2016, from http://CRAN.R-project.org/party/vignettes/party.pdf.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning with applications in R*. New York, NY: Springer.

Konig, I. R., Malley, J. D., Pajevic, S., Weimar, C., Diener, H. C., & Ziegler, A. (2008). Patient-centered yes/no prognosis using learning machines. *International Journal of Data Mining and Bioinformatics*, *2*, 289–341.

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software*, *28*, 340–352.

Kuncel, N. R., Wee, S., Serafin, L., & Hezlett, S. A. (2010). The validity of the Graduate Record Examination for master's and doctoral programs: A meta-analytic investigation. *Educational and Psychological Measurement*, *70*, 340–352.

Li, F., Cohen, A. S., & Shen, L. (2012). Investigating the effect of item position in computer-based tests. *Journal of Educational Measurement*, *49*, 362–379.

Liaw, A., & Weiner, M. (2007). randomforest (R software for random forest). Retrieved June 10, 2016, from http://CRAN.R-project.org/web/packages/randomForest/index.html.

Lin, C. F., Yeh, Y., Hung, Y. H., & Chang, R. I. (2013). Data mining for providing a personalized learning path in creativity: An application of decision trees. *Computers and Education*, *68*, 199–210.

Martin, M. O., & Kelly, D. L. (1996). *Third international mathematics and science study technical report volume 1: Design and development*. Technical Report. Chestnut Hill, MA: Boston College.

Milborrow, S. (2011). rpart.plot: Plot rpart models. An enhanced version of plot.rpart. Retrieved June 10, 2016, from https://cran.r-project.org/web/packages/rpart-plot.

Moses, T. (2012). Relationships of measurement error and prediction error in observed-score regression. *Journal of Educational Measurement*, *49*, 380–398.

Nettles, M. T., & Millett, C. M. (2006). *Three magic letters: Getting to Ph.D*. Baltimore, MD: Johns Hopkins University Press.

Nisbet, R., Elder, J., & Miner, G. (2009). *Handbook of statistical analysis and data mining applications*. Boston, MA: Academic Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., & Cournapeau, D. (2011). scikit-Learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Perna, L. W., Ruby, A., Boruch, R. F., Wang, N., Scull, J., Ahmad, S., & Evans, C. (2014). Moving through MOOCs: Understanding the progression of users in massive open online courses. *Educational Researcher*, *43*, 421–432.

Ramineni, C., & Williamson, D. M. (2013). Automated essay scoring: Psychometric guidelines and practices. *Assessing Writing*, *18*(1), 25–39.

Ridgeway, G. (2014). gbm: Generalized boosted regression modeling. Retrieved June 10, 2016, from http://CRAN.R-project.org/package=gbm.

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*, 601–618.

Schmidt, A. E. (2000). An approximation of a hierarchical logistic regression model used to establish the predictive validity of scores on a nursing licensure exam. *Educational and Psychological Measurement*, *60*, 463–478.

Schulz, E. M., Betebenner, D., & Ahn, M. (2004). Hierarchical logistic regression in course placement. *Journal of Educational Measurement*, *41*, 271–286.

Sheehan, K., & Mislevy, R. J. (1994). *A tree-based analysis of items from an assessment of basic mathematics skills* ETS Research Report No. RR-94-14. Princeton, NJ: Educational Testing Service.

Shieh, Y.-Y., & Fouladi, R. T. (2003). The effect of multicollinearity on multilevel modeling parameter estimates and standard errors. *Educational and Psychological Measurement*, *63*, 951–985.

Stijven, S., Minnebo, W., & Vladislavleva, K. (2011). Separating the wheat from the chaff: On feature selection and feature importance in regression random forests and symbolic regression. In N. Krasnogor (Ed.), *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation* (pp. 623–630). New York, NY: ACM Press.

Stricker, L. J. (2002). *The performance of native speakers of English and ESL speakers on the computer-based TOEFL and GRE general test* (Research Report No. 02-16). Princeton, NJ: Educational Testing Service.

Strobl, C. (2013). Data mining. In T. Little (Ed.), *The Oxford handbook of quantitative methods in psychology* (Vol. 2, pp. 678–700). New York, NY: Oxford University Press.

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, *14*, 323–348.

Subedi, B. R., & Howard, M. (2013). Predicting high school graduation and dropout for at-risk students: A multilevel approach to measure school effectiveness. *Advances in Education*, *2*, 11–17.

Svetnik, V., Liaw, A., Tong, C., & Wang, T. (2004). Application of Breimans random forest to modeling structure-activity relationships of pharmaceutical molecules. In F. Roli, J. Kittler, & T. Windeatt (Eds.), *Multiple classifier systems* (vol. 3077, pp. 334–343). Cagliari, Italy: Springer.

Therneau, T., Atkinson, B., & Ripley, B. (2013). rpart: Recursive partitioning. R package version *4*, 1–3. Retrieved June 10, 2016, from http://CRAN.R-project.org/package=rpart.

Tucker, C., Pursel, B., & Divinsky, A. (2014). Mining student-generated textual data in MOOCs and quantifying their effects on student performance and learning outcomes. *ASEE Computers in Education Journal*, *5*(4), 1–14.

U.S. Department of Education, Office of Educational Technology. (2012). *Enhancing teaching and learning through educational data mining and learning analytics: An issue brief*. Washington, DC: Author.

West, D. M. (2012). *Big data for education: Data mining, data analytics, and web dashboards*. Retrieved June 10, 2016, from http://www.brookings.edu/research/papers/2012/09/04-education-technology-west.

Williams, G. J. (2011). *Data mining with Rattle and R: The art of excavating data for knowledge discovery*. New York, NY: Springer.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., & Zhao, H. (2003). Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics*, *19*, 1636–1643.