Predicting Standardized Test Score Performance:

An Application of Statistical Learning Methods

Collin Kennedy
UC Davis

May 22 April 2021

ABSTRACT

To evaluate and compare the efficacy of various machine learning algorithms and techniques, I train a multitude of machine learning models using a dataset of California school test scores from 1998 and 1999. With predictive performance my primary concern, I ultimately find that LASSO regression and random forests perform most admirably in this continuous predictive setting, whereas classification techniques proved to be relatively poor. While the main goal of this project was to determine which *method* was better in terms of out-of-sample performance, my findings have more relevant implications for policy-making related to standardized testing and schooling.

INTRODUCTION

In 1949, Arthur Samuel, then a professor of electrical engineering at the University of Illinois, joined IBM and began working on their first stored program computer. His primary goal was to construct computers that could learn from their previous experience. Samuel took advantage of the droves of annotated checkers games that distinguished good moves from bad ones, and began designing a program that could improve by having the computer "remember" previous checker moves and compare them with the probability of winning. In 1961, Samuel seized an opportunity to exhibit his work-in-progress program, and challenged the Connecticut state checkers champion to a match, who also just so happened to be ranked 4th in the nation. His program won (McCarthy).

When he finished his checkers program while at IBM, he gave a demonstration of it for none other than Thomas Watson Sr., president and founder of IBM. After seeing what the program was capable of, Watson exclaimed that IBM's stock price would increase by 15 stock points. It did. Watson was witnessing the advent of what Samuel famously described as *Machine Learning* (McCarthy).

Fast forward to today, and statistical machine learning techniques abound. From speech and image recognition, to statistical arbitrage and algorithm-based trading strategies at top hedge funds, statistical learning is front and center (Foote). But perhaps most importantly, statistical learning methods can be used to make important and accurate predictions and classification decisions. Such predictions and classifications may have far-reaching policy implications, such as informing school administrators and districts on optimal levels and allocations of funds, teachers, and resources across schools in respective districts.

To provide some context to the topic and implications of standardized test score performance, consider a recent research brief published by the *Learning Policy Institute,* in which the brief's author- Professor Bruce Baker of Rutgers Graduate School of Education- outlines ways in which schools and school districts with a large proportion of low-income students can still obtain improved student performance, graduation rates, and test scores via increased funding. More specifically, the Professor Baker's study concludes that, "a 21.7% increase in per-pupil spending throughout all 12 school-age years was enough to eliminate the education attainment gap between children from low-income and non-poor families and to raise graduation rates for low-income children by 20 percentage points"("Research Shows"). From an economics standpoint, it's also important to consider the positive externalities of such improvements; such gains can also be seen as reducing the costs associated with welfare programs and crime, while increasing incomes (and by extension, the tax base) which can further contribute to more beneficial and desirable societal outcomes.

Standardized test score performance is also a useful predictor of college acceptance (albeit not as good as grades themselves) and success later in life. In a study published by Raj Chetty, John N. Friedman, and Jonah E. Rockoff in the *American Economic Review,* the economists concluded that students that were assigned to teachers deemed "highly effective" ended up being significantly more likely to attend college as well as earn higher salaries(Chetty et al.).

With that said, it is clear that being able to predict standardized test performance has beneficial implications. By being better-equipped to anticipate the needs of different districts based on student performance, decision-makers, administrators, and policy makers can more effectively address the needs of struggling districts and schools in California and beyond. As such, I sought to apply various machine learning methods, including supervised techniques such as OLS and LASSO regression, K-nearest neighbors, classification and regression trees, random forest, as well as unsupervised statistical learning methods such as principal component analysis to this problem. Because this is by nature a predictive problem, and unsupervised methods do me little good on their own, I instead chose to focus on principal component *regression,* where PCA is performed on the original predictor variables, and then those newly formed principal components are used to train the PCR model. Using data from California schools obtained in 1998 and 1999, I train various machine learning models and compare their predictive ability in order to determine which algorithm and model performs best. More specifically, I explore which machine learning algorithm results in the best prediction of standardized test score performance (in terms of out-of-sample performance) as measured by R^2 and root mean squared error (RMSE). I find that my simplest and most basic supervised model, the OLS regression, performs surprisingly well, with an out-of-sample (OOS) R^2 of 0.78. Beyond OLS, the other more advanced supervised methods, namely PCR-LASSO and LASSO both performed better. However, with the PCR model, the improvement was marginal at best, and based on the convoluted nature and difficult to interpret coefficients, PCR is likely a poor model to consider for use when it comes to predicting standardized test scores. In terms of classification methods, k-nearest neighbors performed the worst by far. The best accuracy rating for a kNN model I trained was about 8%, indicating to me I would be better off classifying individual California schools to percentile rankings based on random chance. Lastly, I considered more advanced and modern classification algorithms, such as classification and regression trees and random forest, and find that random forest performs the best based on mean squared error.

DATA

The cross-sectional data I used to conduct my analysis was collected in 1998 and 1999 from 420 primary school districts (K-6 and K-8) in California. Unsurprisingly, there are 420 observations (one for each school district) and 14 variables. The data contains scores from a standardized test that was given to 5th graders. Among the variables in the data set are relatively self-explanatory factors like `district`, `school`, and `county`. `grades` is a 2-level factor and uses ‘KK-06’ and ‘KK-08’ to denote whether the district is a K-6 or K-8 district, respectively. `students` and `teachers` are numerical variables denoting the number of students and teachers. It is worth pointing out that `teachers` only counts full-time and “full-time equivalent” teachers. `calworks` and `lunch` represent what percentage of the district qualify for CalWorks (income assistance in California) and reduced-priced lunch, and `english` is the percentage of English learners. `computers` denotes the number of computers allocated to each district.

Of most notable interest are `expenditure`, `income`, `read`, and `math`. `expenditure` is a numerical variable that quantifies how much each district spends per student, and `income` is the average level of income in the district (recorded in thousands of US dollars). `read` and `math` are the average scores in the respective sections of the test that was administered to the 5th graders.I also construct five new variables: `score` which is an average of `read` and math`, and `student_teacher_ratio`, which is just `students`/`teachers`. I then create two variables, `decile` and `percentile`, which I constructed for the purposes of being able to apply classification algorithms like k-nearest neighbors to this problem. Lastly, I created an `id` variable, which is just a unique number for each observation that I used to subset the data into training and test sets. Figure 1.

MODEL & RESULTS

To perform my analysis and model selection, I used R. I enlisted the help of external libraries such as *caret* and *gamlr* to perform cross-validation when building my OLS, LASSO and PCR-LASSO models, and used *tree* and *ranger* to create CART and random forest models. The use of *tidyverse* and the multitude of libraries under its umbrella was a no-brainer, as I used *ggplot2* for my visualizations, and made extensive use of the pipe operator for more readable code, which can be found in the appendix.

Prior to any model construction and prediction, I did some data visualization to gain a better understanding of the relationship between variables in the data set. Most notably, I wanted to see if the relationship between a school's/district's expenditures was as highly associated with standardized test scores as Professor Baker's study had found. In R, I aggregated the data by county and calculated average spending (expenditures) and average test score, and superimposed a linear regression line on the plot

Figure 2.

It is apparent that even with this California data, the positive relationship between a given district's level of spending and standardized test scores exists. Given the ostensible appearance of a statistically significant positive relationship, I delved into model building.

i. Ordinary Least Squares Regression

While one of the most fundamental supervised statistical learning methods, OLS regression can be impressively powerful in terms of predictive capability. In R i used *caret*'s train() function to train a linear model and conduct 10-fold cross-validation.

Figure 3.

It appears that *english, income,* and *lunch* are all individually statistically significant at the 5% significance level, and there is overall significance with an F -statistic of 142.3 on 9 and 284 degrees of freedom (p-value < .05). And, there is an (in-sample) R^2 of about .82, so about 82% of the variation in standardized test score performance is explained by the relationship between the regressors and score. The adjusted R^2 , which adds a penalty for the number of predictors in the model, is similarly high, confirming that this model does a good job of explaining the variability in test scores. However, we are concerned with *predicting* test scores, not so much understanding the effect of various variables on standardized test score performance. As such, I then calculated the OOS R^2 using the postResample() function from the caret package. I find the OOS R^2 to be about 78%, which is indicative of good out-of-sample performance.

ii. LASSO Regression

While OLS delivered surprisingly satisfactory out of sample performance, I considered how I could achieve improved performance. First, I considered a Least Absolute Shrinkage and Selection Operator (LASSO) regression. LASSO is a regularization technique that is often employed when there are many variables and model selection may be an overly difficult task given the incredibly high number of possible models. Using LASSO, certain coefficients are shrunk towards 0. This reduces model complexity and can make the coefficients that are left in the model easier to interpret. This introduces a little bias, but there may also be less variance and better predictive performance on out-of-sample data. Mathematically, the objective function that we are minimizing has an added element (constraint) with a tuning parameter λ.

Figure 4.

When λ is 0, we have a model that is equivalent to the OLS estimate, since no shrinkage is applied to the coefficients. However, as the tuning parameter λ increases, more and more bias is introduced into the model. With LASSO, the goal remains the same, with one caveat: Find the model, *and* the value of λ that minimizes the sum of squared errors. In R, I trained the LASSO regression and conducted 10-fold cross validation. In figure 5, shown below, we can see how sensitive root MSE is to the tuning parameter λ.

Figure 5.

Initially, RMSE is noticeably higher, but once λ increases to a certain point, RMSE drops significantly. Similarly to what I did in the case of the OLS model, I then computed out-of-sample R^2 and found it to be about .82. This is a slight improvement over the OLS case, but it is better nonetheless.

iii. Principal Component Regression

I then considered a principal component regression for this problem. PCR can be nice and effective because the linear regression assumption of independence can be better achieved since PCA removes any multicollinearity between the principal components, which in the case of PCR are the regressors . Of course, before I could train the PCR model, I first had to conduct principal component analysis. I dropped all multi-level factor variables (since PCA won't work with them included) and then conducted PCA. The proportion of variance explained can be seen in Figure 5 below:

Figure 6.

Following this step, I then began model building. I converted the output of the PCA (the principal components) into a dataframe, and split it into training and test sets. I then used the cv.gamlr() function to build the PCR model. To evaluate the model (and to better directly compare it to my OLS and LASSO models), I extracted MSE from the model object that was returned from the function, and then calculated OOS R^2 to be about .82. This value is essentially the same as the LASSO model. However, this did not indicate to me that I should be indifferent between using either model to predict standardized test scores. There are many issues with PCR that are difficult to overlook in the context of a highly applied predictive problem. For example, once principal component analysis is applied and our predictors are essentially transformed, the interpretability of the PCR coefficients is impossible, as the principal components used to train the model and derive coefficient estimates have no real-world meaning. While they still contain (the most) valuable information about our dependent variable, test scores, there is no way to interpret them. And, there is also the issue of actually leveraging this model for use in the real-world by administrators and policy makers. With this PCR model, PCA would have to be applied to any additional data that came in from school districts, which could be overly tedious.

Since PCR demonstrated no improvement over the LASSO regression, the justification for such a complex, uninterpretable blackbox method is sorely lacking.

iv. K-Nearest Neighbors

Next, I turned my eyes towards a classification approach to this problem. Standardized test scores are obviously continuous, so I first had to construct a multi-level factor variable from continuous test scores. Here, I had to balance between ensuring that I captured a wide-ranging level of scores, while also making sure to not create a ridiculously "large" categorical variable. I settled on a `percentile` variable, which has 10 levels, and assigned each observation (school) to their respective percentile ranking (i.e., top 10%, 10th percentile, 20th percentile, etc).

The K-nearest neighbors algorithm assigns an observation to a class based on the mode of the set of the K nearest observations (hence "neighbors"). K is therefore a hyperparameter chosen by an individual. I trained a kNN model (K = 5) using the knn() function from the MASS library, and then created a contingency table to evaluate the accuracy of the model. The sum of the diagonal of the table gives the accuracy rating, and can be seen in figure 7 below along with the contingency table I created.

Figure 7.

Clearly, kNN does not perform very well for this problem. With an accuracy rating of about 8%, this is worse than if schools were classified into different decile rankings solely based on chance alone (i.e., if schools were classified based on the roll of a 10-sided die). Because of its weak performance, I moved on to considering other nonparametric methods that may be better suited for this problem.

v. Classification and Regression Tree (CART) & Random Forest

Here, I consider two nonparametric methods for prediction: CART and random forest. Both methods can be useful in regression and binary classification settings. Both approaches divide the predictor space into distinct regions, based on the minimization of the error, and each decision is a node, where the final prediction is referred to as a leaf node. Where they differ is that a random forest tree better avoids overfitting by fitting a CART with a bootstrapped sample, and each split uses a randomly chosen subset of predictors. Then, the average of each of the B predicted probabilities is used to make the final prediction.

In R, I created a for loop to fit 10 different CART, LASSO regression, and Random forest models. I stored the 10 mean squared error calculations of each model in a list, and then produced a boxplot to visualize each model's MSE. I included the LASSO regression here for comparison, because up until this point, it had been performing the best, and was my standard baseline for comparison. The boxplot can be seen below in Figure 8.

Figure 8.

It appears that both the LASSO regression and random forest tree are performing similarly well. The CART, meanwhile, has an average MSE (average over the 10 CART models constructed) that is about 40% greater than the respective MSEs of both the LASSO regressions and random forest models.

Overall, the LASSO regression model and the random forest model(s) performed the best in terms of out of sample performance. Meanwhile, k-nearest neighbors proved to be almost disastrous. With an accuracy rating of roughly 8%, I could achieve better prediction of decile ranking of a school in terms of standardized test score performance by having a blind-folded monkey randomly point his (or her) banana at any of the given possible levels of the `percentile` variable. This was inherently not a classification problem, and that is clearly reflected in kNN's measly performance. The LASSO regression had an OOS R^2 of about 82%, about on par with the PCR model and slightly higher than the standard OLS model. Because the coefficients of PCR are practically impossible to interpret, yet the OOS R^2 for both models are roughly the same, LASSO is definitely a top pick. Using a similar argument, I would say that LASSO is probably the best model to use overall, as its closest competitor in this case, the random forest tree, is essentially a machine learning black box that can be difficult to understand. The LASSO regression has coefficients that are actually interpretable, and is likely more easily understood by people that would be using its predictions to make important decisions.

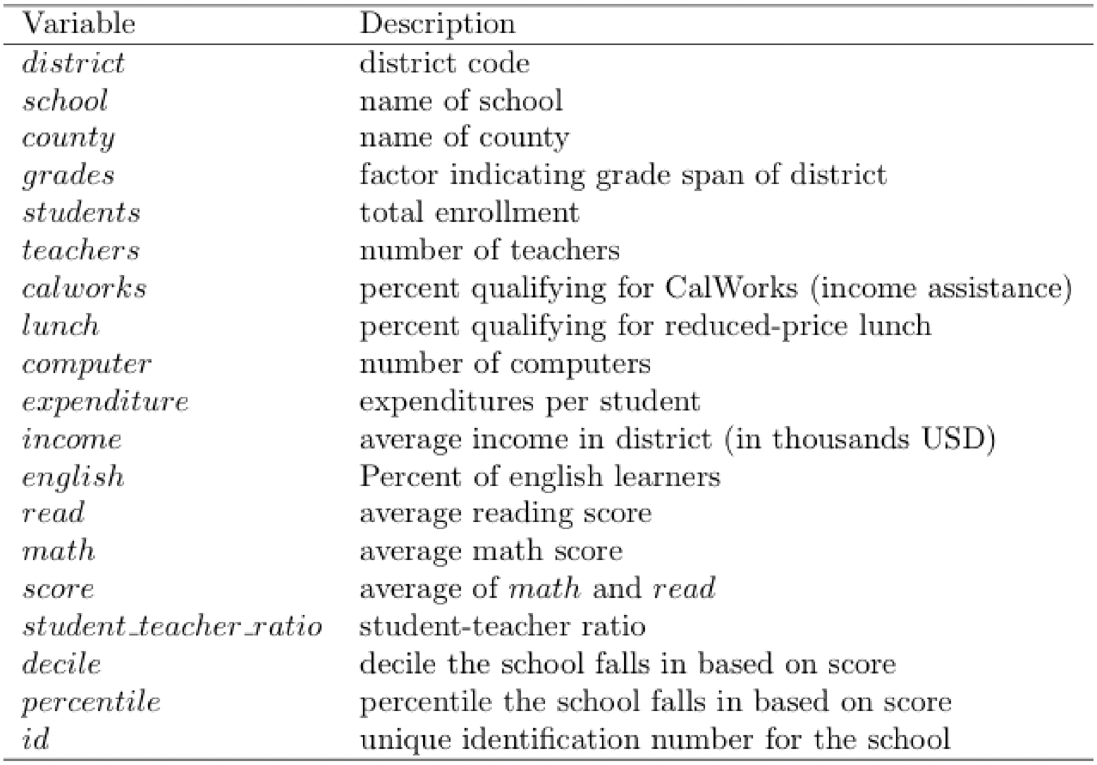
CONCLUSION

By developing statistical learning models to accurately predict standardized test score performance, school districts across California and other regions can be better equipped to identify areas of much needed improvement in their respective schools.

Some organizations are already implementing statistical frameworks to identify areas of improvement in schools. For example, The National High School Center at the American Institutes for Research (AIR) has developed a system to detect students that are at greater risk of not graduating high school, and the University of Chicago has developed the Risk and Opportunity Framework, which serves as an early warning indicator for educators to identify and predict how likely students are likely to succeed in high school and graduate based on performance in 8th grade. These applications of prediction in an education setting have already proven to be effective; the AIR conducted a trial to evaluate the efficacy of their system and found it significantly reduced the number of failures among students(“The Importance”).

The outlook for the application of statistical learning methods in improving educational outcomes is indeed promising. In many ways, that outlook is already becoming a reality.

APPENDIX

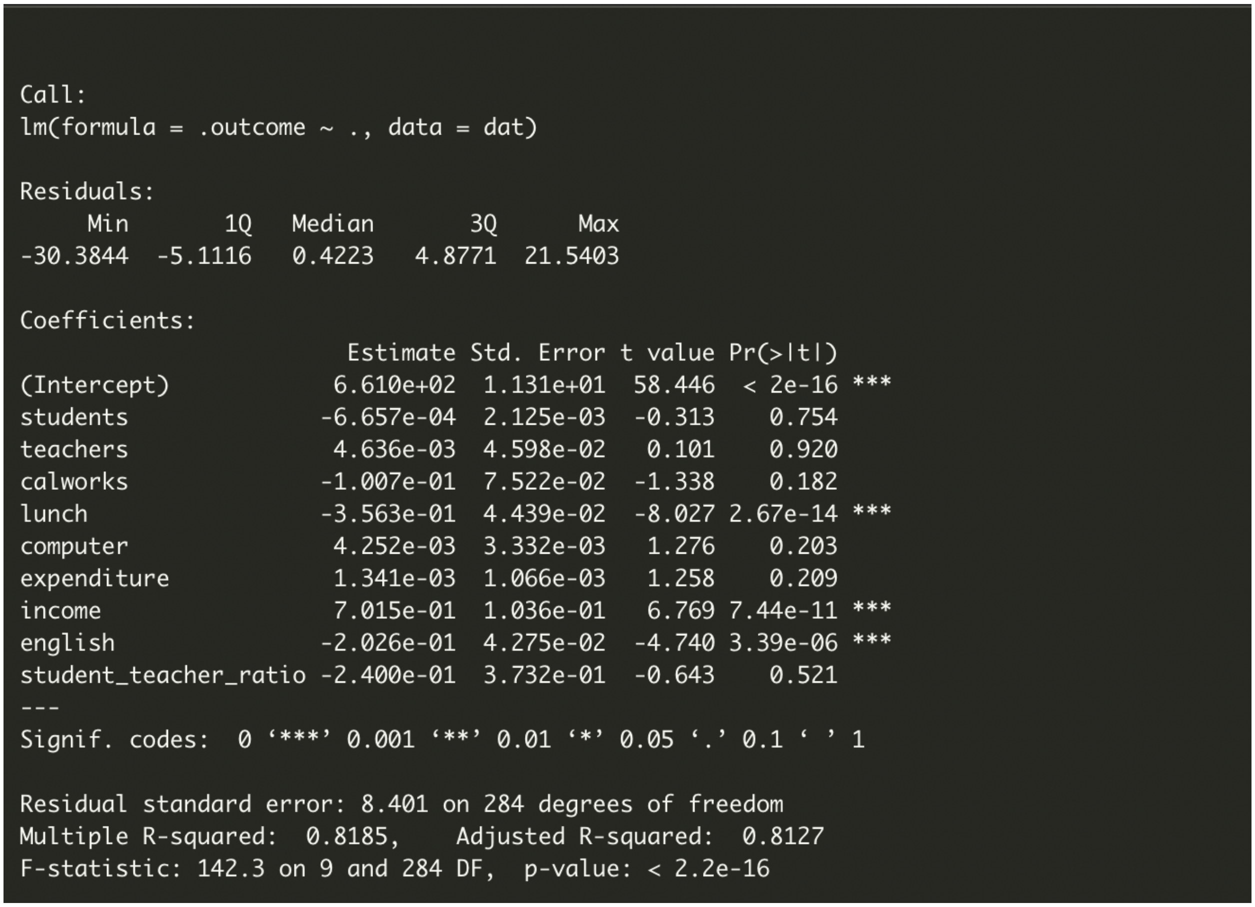


Variable	Description
district	district code
school	name of school
county	name of county
grades	factor indicating grade span of district
students	total enrollment
teachers	number of teachers
calworks	percent qualifying for CalWorks (income assistance)
lunch	percent qualifying for reduced-price lunch
computer	number of computers
expenditure	expenditures per student
income	average income in district (in thousands USD)
english	Percent of english learners
read	average reading score
math	average math score
score	average of *math* and *read*
student_teacher_ratio	student-teacher ratio
decile	decile the school falls in based on score
percentile	percentile the school falls in based on score
id	unique identification number for the school

Table 1: Variables in the analyses

Figure 1.

Figure 2.



```
Call:
lm(formula = .outcome ~ ., data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-30.3844  -5.1116   0.4223   4.8771  21.5403 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)            6.610e+02  1.131e+01  58.446  < 2e-16 ***
students              -6.657e-04  2.125e-03  -0.313    0.754    
teachers               4.636e-03  4.598e-02   0.101    0.920    
calworks              -1.007e-01  7.522e-02  -1.338    0.182    
lunch                 -3.563e-01  4.439e-02  -8.027 2.67e-14 ***
computer               4.252e-03  3.332e-03   1.276    0.203    
expenditure            1.341e-03  1.066e-03   1.258    0.209    
income                 7.015e-01  1.036e-01   6.769 7.44e-11 ***
english               -2.026e-01  4.275e-02  -4.740 3.39e-06 ***
student_teacher_ratio -2.400e-01  3.732e-01  -0.643    0.521    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.401 on 284 degrees of freedom
Multiple R-squared:  0.8185,	Adjusted R-squared:  0.8127 
F-statistic: 142.3 on 9 and 284 DF,  p-value: < 2.2e-16
```

Figure 3.

$$\sum_{i=1}^{n}(y_i - \sum_{j} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Figure 4.

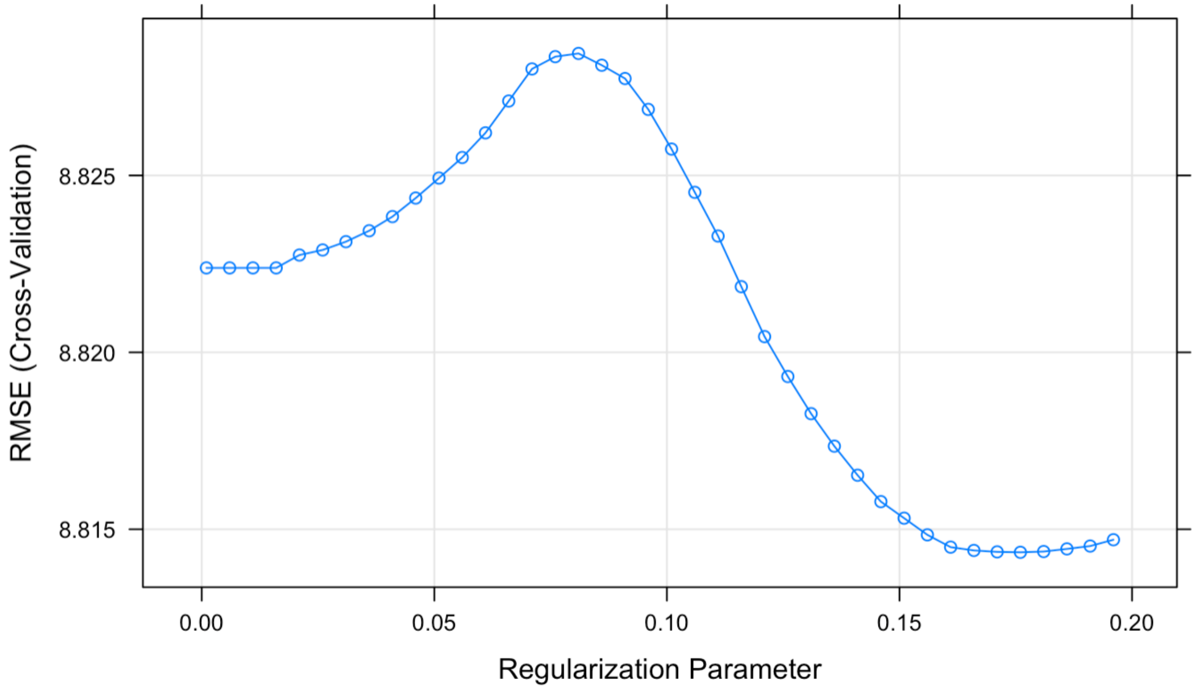
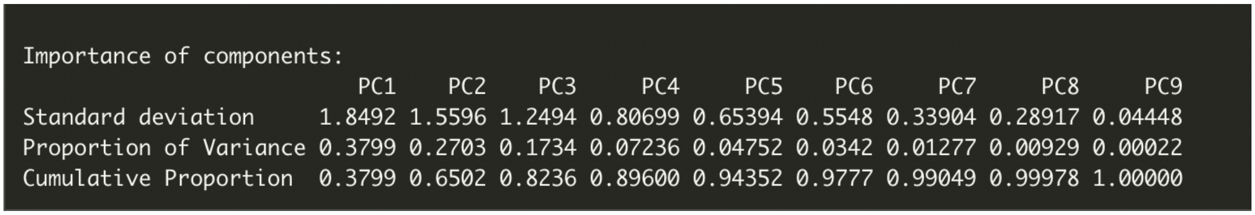
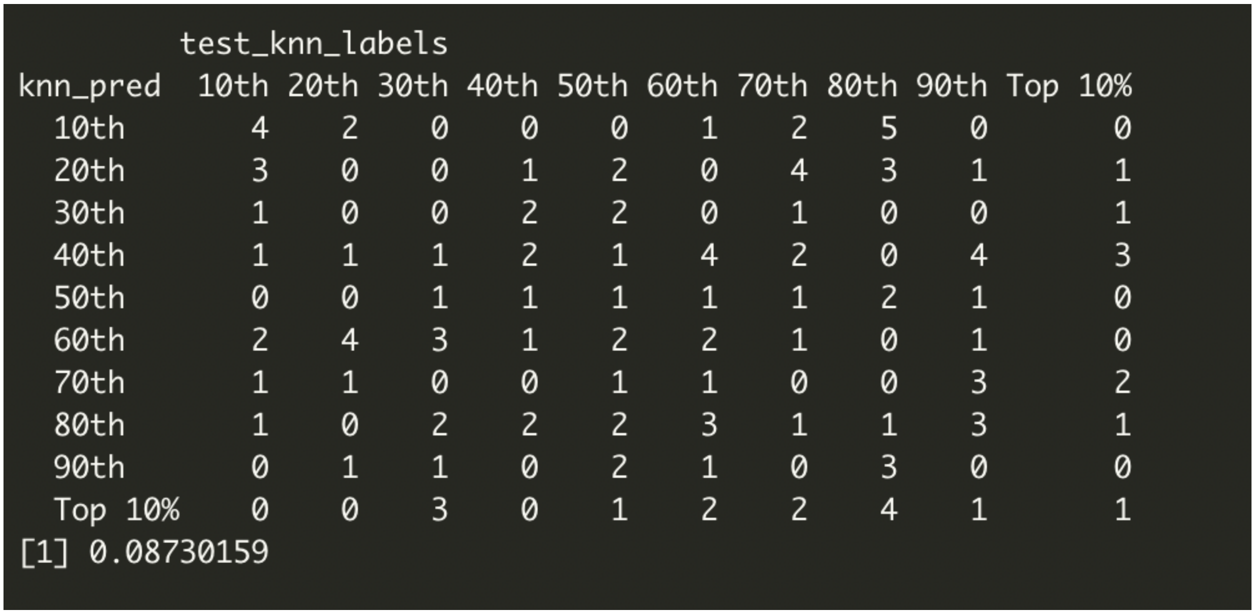


Figure 5.



```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5    PC6     PC7     PC8     PC9
Standard deviation     1.8492 1.5596 1.2494 0.80699 0.65394 0.5548 0.33904 0.28917 0.04448
Proportion of Variance 0.3799 0.2703 0.1734 0.07236 0.04752 0.0342 0.01277 0.00929 0.00022
Cumulative Proportion  0.3799 0.6502 0.8236 0.89600 0.94352 0.9777 0.99049 0.99978 1.00000
```

Figure 6.



```
          test_knn_labels
knn_pred   10th 20th 30th 40th 50th 60th 70th 80th 90th Top 10%
  10th        4    2    0    0    0    1    2    5    0       0
  20th        3    0    0    1    2    0    4    3    1       1
  30th        1    0    0    2    2    0    1    0    0       1
  40th        1    1    1    2    1    4    2    0    4       3
  50th        0    0    1    1    1    1    1    2    1       0
  60th        2    4    3    1    2    2    1    0    1       0
  70th        1    1    0    0    1    1    0    0    3       2
  80th        1    0    2    2    2    3    1    1    3       1
  90th        0    1    1    0    2    1    0    3    0       0
  Top 10%     0    0    3    0    1    2    2    4    1       1
[1] 0.08730159
```

Figure 7.

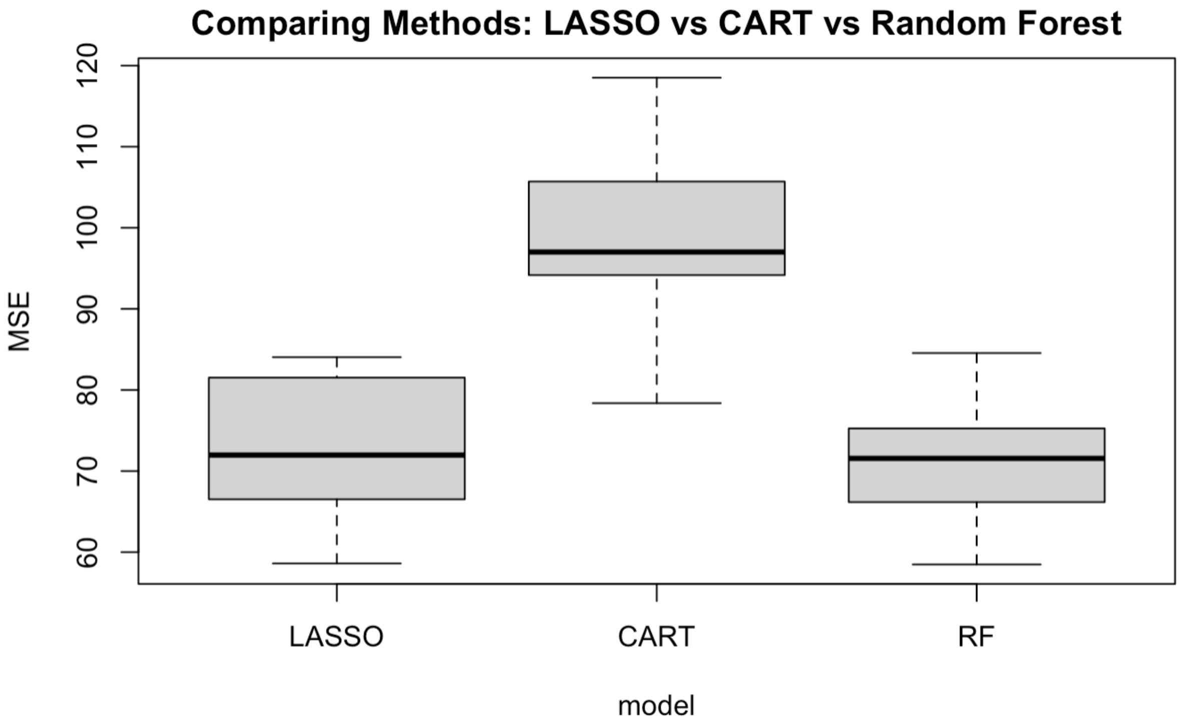


Figure 8.

<u>REFERENCES</u>

Chetty, Raj, et al. "Measuring the Impacts of Teachers II: Teacher Value-Added and Student Outcomes in Adulthood†." *Http://Dx.doi.org*, American Economic Review, Sept. 2014, pubs.aeaweb.org/doi/pdf/10.1257/aer.104.9.2633.

Foote, Keith D. "A Brief History of Machine Learning." *DATAVERSITY*, 13 Mar. 2019, www.dataversity.net/a-brief-history-of-machine-learning/.

"The Importance of Grades." *Uei.uchicago.edu*, University of Chicago- Urban Education Institute, uei.uchicago.edu/sites/default/files/documents/UEI%202017%20New%20Knowledge%20-%20The%20Importance%20of%20Grades.pdf.

McCarthy, John. "Arthur Samuel: Pioneer in Machine Learning." *Infolab.stanford.edu*, Stanford Computer Science Computer History Display, infolab.stanford.edu/pub/voy/museum/samuel.html.

"Research Shows That When It Comes to Student Achievement, Money Matters." *Learning Policy Institute*, Learning Policy Institute, 2018, learningpolicyinstitute.org/press-release/research-shows-student-achievement-money-matters.