FinalReport

Meghan Myles and Owen Fiore

Student Exam Excellence

Abstract

Standardized tests have been used to try and evaluate how much each student has learned in a classroom. However, it is possible that predetermined factors related to a student's socioeconomic status play a significant role in how student's learn and how well they perform on standardized tests. It is important to understand that although standardized tests are meant to level the playing field between students, there are many elements that can contribute to student performance on standardized tests.

Introduction

It is important to be able to measure students' academic performance, as such measurements can help guide decisions about best educational practices at every level. They can help to identify individual students who are performing poorly in order to give them more individualized instruction. They can identify districts which are underperforming in order to better invest educational resources. They can identify subject areas in which students are consistently underperforming in order to tailor teaching methods. One method of evaluating students' academic performance is by investigating their scores on standardized exams.

It is desirable to explore factors influencing standardized test scores to determine if standardized tests are performing as they should be. Ideally we would like to see that standardized test scores are poorly correlated with socioeconomic factors and very strongly correlated with how much students study for them. This would be indicative that standardized test scores and thus learning are not impacted very much by prior factors and that students of all backgrounds have the chance to succeed.

There have been a number of investigations into the interactions between socioeconomic factors and student academic performance. According to the NIH, around half of people who grew up in a high-income family have a bachelor's degree by the age of 25, while only about 10% of

people from low-income families have a bachelor's degree by 25. In addition, there tend to be fewer AP classes offered in schools which traditionally serve low-income and minority students (National Institutes of Health). In addition, on certain standardized tests, some groups tend to consistently outperform others. On the SAT II writing exam, the average white student outperformed the average student of color (Thomas 2004). A University of South Florida study showed that students' SAT and ACT scores were not predictive of their college academic performance (Micceri 2010). In a 2009 study, Black and latino students performed lower on the MCAT, on average, than white students (Davis 2013). It is clear that standardized test scores are not necessarily only indicative of academic success - it seems that they also capture variation introduced by socioeconomic factors.

We will examine a dataset detailing students' standardized test scores, as well as measures for several socioeconomic factors and student dedication to academics. In doing so, we will be able to determine whether or not this standardized test is appropriate for measuring true student achievement.

Data Description:

Here is a link to the kaggla data: https://www.kaggle.com/datasets/desalegngeb/students-exam-scores

- Gender: Gender of the student (male or female)
- EthnicGroup: Ethnic group of the student (nominal groups from A to E)
- ParentEduc: Parent education background: (ordinal groups from some high school to master's)
- LunchType: Cost of school lunch (free/reduced or standard)
- TestPrep: Whether the student completed a test preparation course (none or completed)
- ParentMaritalStatus: Student's parent's marital status(nominal groups: married, single, widowed, divorced)
- PracticeSport: How often a student plays a sport (ordinal groups from never, sometimes, regularly)
- IsFirstChild: If the student is the eldest child in their family (yes or no)
- NrSibling: number of siblings the student has
- TransportMeans: how the student gets to school (schoolbus or private)
- WklyStudyHours: how much the student self-studies (ordinal groups from 0-5, 5-10, and 10+ hours)

MathScore, ReadingScore, WritingScore are the student's math, reading, and writing scores respectively and serve as the response variable in this dataset. Scores range from 0 to 100 with 0 indicating low performance and 100 indicating perfect.

We will treat the variables Gender, EthnicGroup, ParentEduc, ParentMaritalStatus, PracticeSport, NrSibling, and TransportMeans as indicators of students' socioeconomic status. We

will treat the variables TestPrep and WklyStudyHours as indicators of students' academic dedication.

As this dataset is fictional, it is impossible to perform any additional research on the background of the town this data came from or provide more context.

Goal

Our goal is to determine whether or not standardized tests are performing well in measuring academic performance. Our research questions are as follows:

Do math, reading, and writing scores on this standardized test correlate with each other? Do performances on the tests correlate with socioeconomic factors, such as highest parental education level or receiving a free/reduced lunch? Do performances on the tests correlate with indicators of academic commitment, such as studying for many hours weekly and completing test preparation courses? Do test scores vary significantly across gender and ethnic groups? In all, does it appear that each of these tests are measuring true academic performance, or are they heavily influenced by other factors?

By regressing test scores on the other variables available in this dataset, we hope to be able to determine answers to these questions. We will determine whether standardized test performance is more highly correlated with socioeconomic factors or academic dedication. This will help us to determine whether or not the tests are appropriate measures of academic performance.

Methods

Generalized Linear Model

In this paper we will test out the use of several different models of varying complexities to try and best fit our data, starting with a linear model. Regression diagnostics showed that the model was over-predicting the achievement of students who should be doing really well. Thus we decided to implement a Generalized Linear Model to try and remedy this. A generalized linear model has a random component, systematic component, and a link function.

The random component is a probability density function from an exponential family:

$f(y;\theta,\phi) = \exp\frac{y\theta - b(\theta)}{a(\phi)+c(y,\phi)}$ where θ and $\phi > 0$ are scalar parameters, and a, b, c, are known functions. The form of a Gaussian distribution with an identity link is:

$f(y\|\mu,\sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)}}\exp(-\frac{(y-\mu)^2}{2\sigma^2})$. Where μ is the mean of the distribution and σ^2 is the variance

The systematic component is the main part of the model and is a generalized form of a linear model such that for the systematic component: η_i we have:

$\eta_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}$

To obtain this part, we implemented stepwise selection through the `step` R function in both directions, allowing both forward and backwards selection. Forward or backwards selection is a variable selection algorithm that looks for the most significant predictors and returns a model when there cannot be any additional significant variables added. The algorithm returned that the following parameters should be included in the model: `Male`, `ParentEduc`, `LunchDiscount`, `TestPrep`, `PracticeSport`, `IsFirstChild`, `WklyStudyHours`, and all variables of the form `EthnicGroup`. Thus our model looks like:

$\eta_i = \beta_0 + \beta_1 Male + \beta_2 ParentEduc + ... + \beta_{11} EthnicGroupE$

The final part of a GLIM is the link function which has the form:

$g(\mu_i) = \eta_i$

Where $g()$ is a continuous function that maps the mean response μ_i to its systematic component η_i. Several link functions were considered, but the one that showed the best results when utilizing a normal quantile-quantile plot was a Gaussian family with identity link, so that will be the GLIM presented throughout the rest of the paper. Additionally, a Gaussian family with identity link is highly interpretable, as the coefficients can be interpreted as how they appear in the model.

Random Forests

In addition to the linear model methods described above, we wanted to implement more complex methods through more sophisticated models. Random Forests are an ensemble learning method that use a collection of decision trees to create optimal predictions. Decision trees typically work using the CART algorithm, which is a greedy algorithm that minimizes the number of incorrectly classified cases, typically measured by the Gini index of a node. Eventually stopping conditions will be met based on a number of potential factors, but the one used in this paper is minimum node size. As random forests are made up of a number of decision trees, the number of trees is a hyper-parameter in the input of the random forest. As we have multiple hyperparameters that we want to control: number of trees and minimum node size, we will implement grid search cross validation in order to iterate through possible combinations of hyperparameters to find the optimal hyperpaperameters. After running the search, we concluded that the optimal number of trees was 900 and that we should have a minimum of 25 observations in each node. Additionally, we will need to analyze how important each variable was in building the random forests. Random forests are a collection of trees, and tree based methods work by splitting at the most decisive points to try and maximize the separation between children nodes. Thus, variables that are indicated to be important are significant predictors in `AggregateScore`, as the predicted score will change decently based on whether

an observation has that particular feature or not. While we hope to see that socio-economic factors do not play a large role in performance, we know from prior reading that is not likely to be the case. Nonetheless, this analysis is important and aims to motivate further questions about the quality of standardized tests.

Results

```
df <- read.csv("DataToModel.csv")

predictor_variables <- df[, c("MathScore", "ReadingScore", "WritingScore", "AggregateScore
cor(predictor_variables)
```

```
               MathScore ReadingScore WritingScore AggregateScore
MathScore      1.0000000    0.8189865    0.8085328      0.9205491
ReadingScore   0.8189865    1.0000000    0.9526215      0.9694976
WritingScore   0.8085328    0.9526215    1.0000000      0.9665601
AggregateScore 0.9205491    0.9694976    0.9665601      1.0000000
```

We have four predictor variables in our dataset: `MathScore`, `ReadingScore`, `WritingScore`, and `AggregateScore` which is the sum of the previous three. The correlation results show that the scores are generally very highly correlated and thus it seems reasonable for this section to only use `AggregateScore`, as the results should generalize well to the other three scores. This will help to keep the results concise as any results that can be applied to `AggregateScore` should be able to be applied to the other variables. Logically, it also does not seem reasonable that many of the predictors such as `LunchType` would be good at predicting `MathScore` but not `ReadingScore`.

Now that we have chose `AggregateScore` as our predictor, we can split our data into 80% training and 20% testing.

```
set.seed(123457)
train.prop <- 0.80
strats <- df$AggregateScore
rr <- split(1:length(strats), strats)
idx <- sort(as.numeric(unlist(sapply(rr,
        function(x) sample(x, length(x)*train.prop)))))
df.train <- df[idx, ]
df.test <- df[-idx, ]
```

Linear Model Results

Now, we are ready to implement a linear model. Several simpler models were considered without exponential families, but were excluded for the purpose of keeping the Results section concise, but extended work and methodology can be found in the Supplementary Files folder under "Models.Rmd".

We can see how the fit is with all predictor in the model.

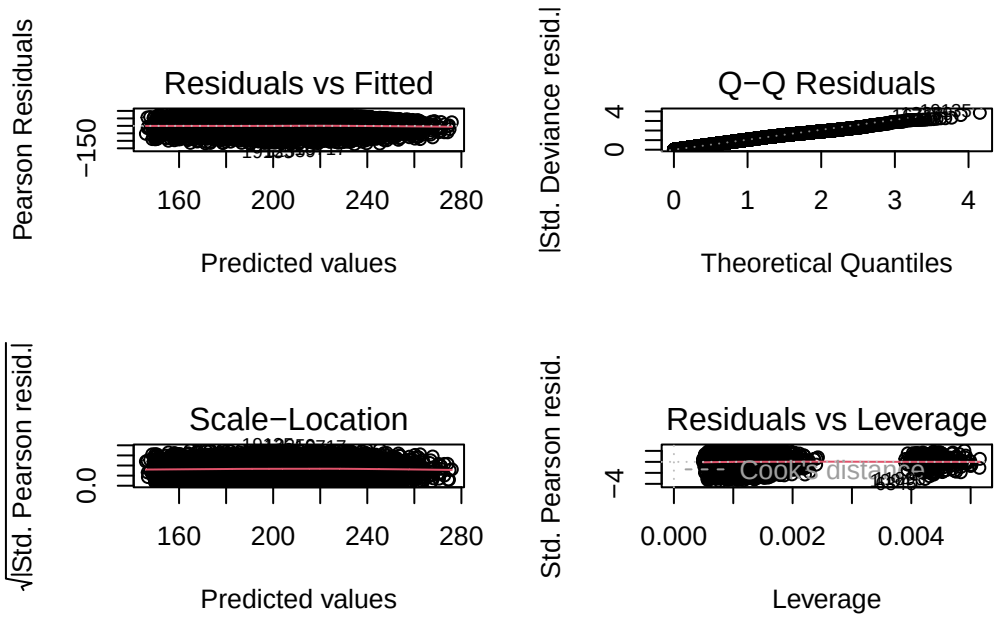
```
predictors <- c("Male", "ParentEduc", "LunchDiscount", "TestPrep", "PracticeSport", "IsFir

all <- glm(AggregateScore ~ Male + ParentEduc + LunchDiscount + TestPrep + PracticeSport +
summary(all)
```

```
Call:
glm(formula = AggregateScore ~ Male + ParentEduc + LunchDiscount +
    TestPrep + PracticeSport + IsFirstChild + NrSiblings + PrivateTransport +
    WklyStudyHours + EthnicGroupA + EthnicGroupB + EthnicGroupC +
    EthnicGroupD + EthnicGroupE + ParentDivorced + ParentMarried +
    ParentSingle + ParentWidowed, family = gaussian(link = "identity"),
    data = df.train)

Coefficients: (2 not defined because of singularities)
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      202.5949     2.7631  73.321  < 2e-16 ***
Male             -11.4011     0.5950 -19.163  < 2e-16 ***
ParentEduc         6.6454     0.1994  33.329  < 2e-16 ***
LunchDiscount    -29.4773     0.6234 -47.288  < 2e-16 ***
TestPrep          20.3328     0.6248  32.543  < 2e-16 ***
PracticeSport      3.5921     0.4463   8.049 8.96e-16 ***
IsFirstChild       0.9390     0.6254   1.501    0.133
NrSiblings         0.1501     0.2073   0.724    0.469
PrivateTransport  -0.1088     0.6039  -0.180    0.857
WklyStudyHours     0.5714     0.3885   1.471    0.141
EthnicGroupA     -26.3222     1.3425 -19.606  < 2e-16 ***
EthnicGroupB     -25.8734     1.0288 -25.148  < 2e-16 ***
EthnicGroupC     -22.6545     0.9491 -23.869  < 2e-16 ***
EthnicGroupD     -13.0246     0.9831 -13.249  < 2e-16 ***
EthnicGroupE           NA         NA      NA       NA
ParentDivorced    -0.4883     2.2864  -0.214    0.831
ParentMarried     -1.0581     2.2052  -0.480    0.631
```

```
ParentSingle      -1.5822     2.2535  -0.702    0.483
ParentWidowed          NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 1350.116)

    Null deviance: 28262312  on 15283  degrees of freedom
Residual deviance: 20612220  on 15267  degrees of freedom
AIC: 153559

Number of Fisher Scoring iterations: 2
```

We see there are too many parameters in the model, as several such as `IsFirstChild` and `NrSiblings` are not significant. Additionally, we do not to include some variables that were one-hot encoded such as `EthnicGroupE` and `ParentWidowed`, as if an observation is not in Ethnic Groups A-D then it is known that person must have come from Ethnic Group E. There is almost certainly multicollinearity, which will have to be resolved later.

```r
par(mfrow = c(2,2))
plot(all)
```



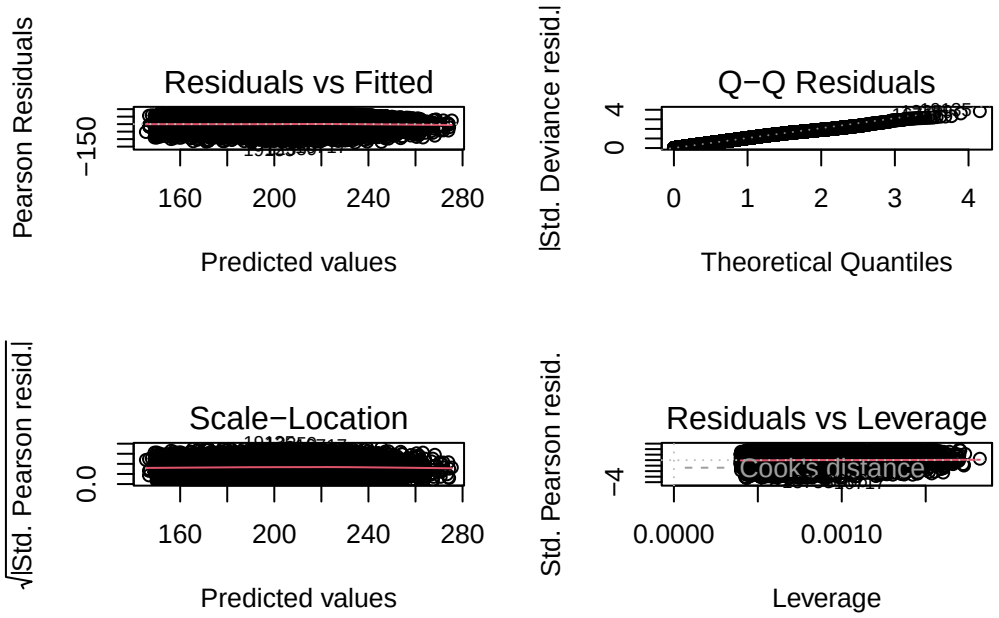
Despite potential problems with too many variables, we see strong results, the residuals appear to be normally distributed, and although the normal q-q plot shows that some values are underfitted, it is reasonable given that we are working with standardized test data where there is a maximum score (300 as this is `AggregateScore` we are modeling) and thus even the students who have the highest probability to score extremely high are bounded by an upper score and thus the observed values are lower than the theoretical. Next, we will use step selection to remove poor indicating variables, while hoping to attain similar residual plots.

```r
step_model <- step(all, direction = "both", trace = 0)
summary(step_model)
```

```
Call:
glm(formula = AggregateScore ~ Male + ParentEduc + LunchDiscount +
    TestPrep + PracticeSport + IsFirstChild + WklyStudyHours +
    EthnicGroupA + EthnicGroupB + EthnicGroupC + EthnicGroupD,
    family = gaussian(link = "identity"), data = df.train)

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    201.8452     1.6488 122.422  < 2e-16 ***
Male           -11.4038     0.5946 -19.178  < 2e-16 ***
ParentEduc       6.6478     0.1993  33.351  < 2e-16 ***
LunchDiscount  -29.4815     0.6233 -47.301  < 2e-16 ***
TestPrep        20.3223     0.6246  32.536  < 2e-16 ***
PracticeSport    3.5873     0.4462   8.040 9.66e-16 ***
IsFirstChild     0.8918     0.6207   1.437    0.151
WklyStudyHours   0.5765     0.3883   1.484    0.138
EthnicGroupA   -26.3258     1.3420 -19.616  < 2e-16 ***
EthnicGroupB   -25.8893     1.0286 -25.169  < 2e-16 ***
EthnicGroupC   -22.6609     0.9488 -23.883  < 2e-16 ***
EthnicGroupD   -13.0288     0.9828 -13.257  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 1349.867)

    Null deviance: 28262312  on 15283  degrees of freedom
Residual deviance: 20615168  on 15272  degrees of freedom
AIC: 153552

Number of Fisher Scoring iterations: 2
```

The step algorithm was able to remove many of the insignificant predictors and able to cut down the list of predictors to only the most important. It is worth noting that although there are some variables that are technically not significant at $\alpha = 0.05$, the coefficient estimates and standard errors show that the variables are still adding value.

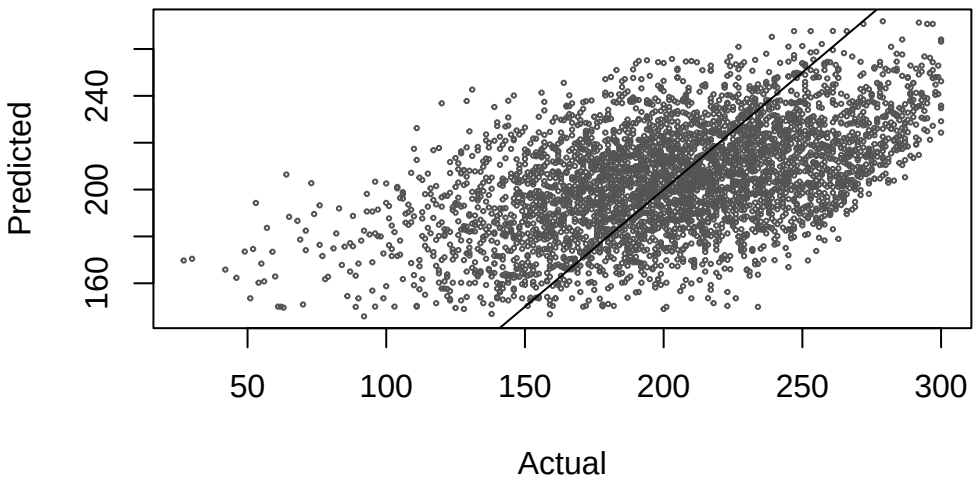
```
par(mfrow = c(2,2))
plot(step_model)
```



As we hoped to see, the residual plots all still appear to be strong, indicating that there are no outlier points exerting undue influence on the model, and no particularly high leverage points as well.

We can now see how well we did by validating on the test dataset.

```
test_predictions <- predict(step_model, newdata = df.test, type = "link")
test_results <- data.frame(Actual = df.test$AggregateScore, Predicted = test_predictions)
plot(test_results$Actual, test_results$Predicted, col="grey33", cex=0.3, xlab="Actual", yl
abline(0,1)
```



While there was error, it appears to be normally distributed around the y=x line, and helps to show that the model fit is good. Now we need to see if the step selection reduced multicollinearity and if there were any issues with variables having a high VIF.

```
car::vif(step_model)
```

```
        Male     ParentEduc  LunchDiscount       TestPrep  PracticeSport
    1.000506       1.000538       1.000517       1.000465       1.000544
IsFirstChild WklyStudyHours   EthnicGroupA   EthnicGroupB   EthnicGroupC
    1.000412       1.000772       1.416668       1.944399       2.220065
EthnicGroupD
    2.099234
```

There is no evidence to suggest there is multicollinearity, as many values are very close to 1, with the exception of the Ethnic Groups which are one-hot encoded and thus dependent on each other. These results show that the step function did a good job of selecting the correct variables to be used.

```
mse <- mean((test_results$Actual - test_results$Predicted)^2)
mse
```

```
[1] 1501.613
```

The mean squared error of the GLIM is 1501.613. We can now compare these results to what we get when implementing a random forest.

Random Forest Results

```
set.seed(1)
library(ranger)
rf_step <- ranger(AggregateScore ~ Male + ParentEduc + LunchDiscount + TestPrep + Practice
rf_step
```

```
Ranger result

Call:
 ranger(AggregateScore ~ Male + ParentEduc + LunchDiscount + TestPrep +      PracticeSport +

Type:                             Regression
Number of trees:                  500
Sample size:                      15284
Number of independent variables:  12
Mtry:                             3
Target node size:                 5
Variable importance mode:         impurity
Splitrule:                        variance
OOB prediction error (MSE):       1372.105
R squared (OOB):                  0.2580269
```

We can run a random forest with many default parameters that use 500 trees and a target node size of 5, but we also expect that we can improve results by changing hyperparameters. Using the methods described above we can use a brute force algorithm to find the best combinations of hyperparameters in hopes of finding something with a lower Out Of Bag prediction error, which is represented by mean squared error.

```
set.seed(1)
library(ranger)

# Define your custom function for training a random forest
train_rf <- function(num.trees, min.node.size) {
```

```
  # Create the random forest model
  rf_model <- ranger(AggregateScore ~ Male + ParentEduc + LunchDiscount + TestPrep + Pract
                     data = df.train,
                     importance = "impurity",
                     num.trees = num.trees,
                     min.node.size = min.node.size)

  # Calculate OOB error (MSE)
  oob_error <- sqrt(rf_model$prediction.error)

  # Return the trained model and OOB error
  return(list(model = rf_model, oob_error = oob_error))
}

# Example usage:
# Replace 'your_data' with your actual dataset and 'your_target_col' with your target vari
# Set the hyperparameter values you want to try
num_trees_values <- c(300, 500, 700, 900)
min_node_size_values <- c(20, 25, 30, 35)

# Initialize variables to keep track of the best model and its OOB error
best_model <- NULL
best_oob_error <- Inf

# Perform grid search
for (num_trees in num_trees_values) {
  for (min_node_size in min_node_size_values) {
    result <- train_rf(num_trees, min_node_size)
    model <- result$model
    oob_error <- result$oob_error

    # Check if the current model has a lower OOB error
    if (oob_error < best_oob_error) {
      best_model <- model
      best_oob_error <- oob_error
    }
  }
}

# The best model is stored in 'best_model' with the lowest OOB error
print(best_model)
```

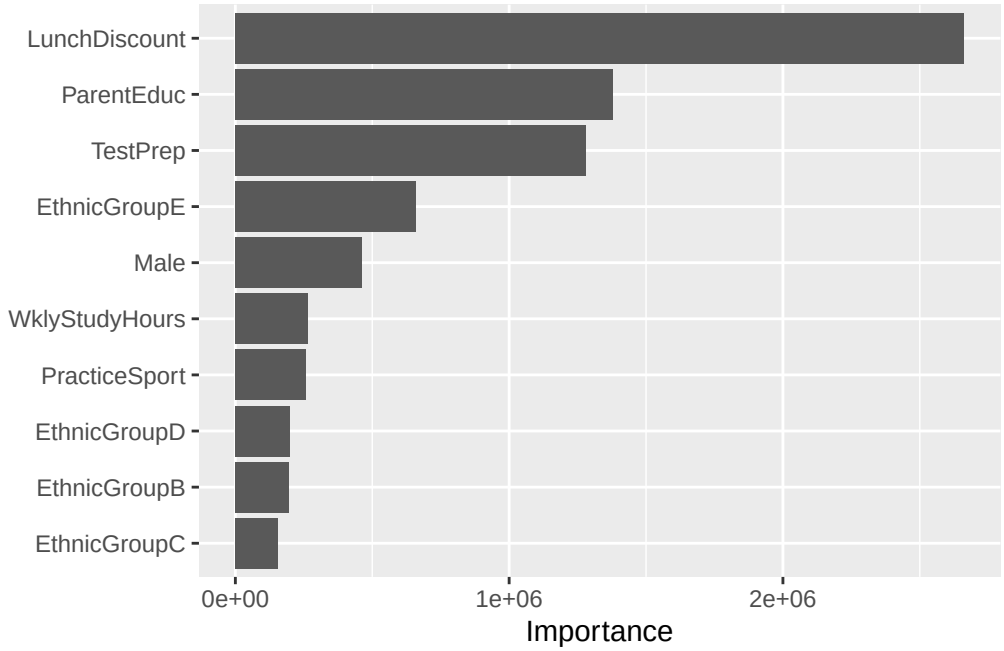
```
Ranger result

Call:
 ranger(AggregateScore ~ Male + ParentEduc + LunchDiscount + TestPrep +      PracticeSport +

Type:                             Regression
Number of trees:                  900
Sample size:                      15284
Number of independent variables:  12
Mtry:                             3
Target node size:                 25
Variable importance mode:         impurity
Splitrule:                        variance
OOB prediction error (MSE):       1368.377
R squared (OOB):                  0.2600428
```

We see that the best model has 900 trees and has a target node size of 25 rather than 5. Despite these changes, we unfortunately do not see a huge change in the mean squared error, as the improvement was quite small. Nonetheless, the mean squared error from this model was 1368.4 which is an improvement over the error produced by the GLIM, which was 1501.6. In order to get a better understanding for the effectiveness of the random forest, it is helpful to look at variable importance. The graph below summarizes the importance of each variable that was included in the random forest.

```
library(vip)
```

```
Attaching package: 'vip'

The following object is masked from 'package:utils':

    vi
```

```
importance <- vi(best_model)
vip(importance)
```



We see that `LunchDiscount` is by far the most important variable, followed by `ParentEduc` and then `TestPrep`, indicating that the largest factor in predicting student performance in a random forest is based on whether or not a school qualifies for free or reduced lunch. Having reduced or free lunch is a measure of the income of the student's family, and thus it is reasonable to conclude that family income is a major factor in determining `AggregateScore`. We can confirm these results by viewing the output of the GLIM, and after doing so we see that the coefficient on `LunchDiscount` is -29.5, which means that we expect to see, on average a student that gets free/reduced lunch score 29.5 points lower than somebody who does not. ## Summary and Conclusion

References

Davis, Dwight MD; Dorsey, J. Kevin MD, PhD; Franks, Ronald D. MD; Sackett, Paul R. PhD; Searcy, Cynthia A. PhD; Zhao, Xiaohui PhD. Do Racial and Ethnic Group Differences in Performance on the MCAT Exam Reflect Test Bias?. Academic Medicine 88(5):p 593-602, May 2013. | DOI: 10.1097/ACM.0b013e318286803a

"Individuals from Disadvantaged Backgrounds." National Institutes of Health, U.S. Department of Health and Human Services, extramural-diversity.nih.gov/diversity-matters/disadvantaged-backgrounds.

Micceri, Theodore. "Assessing the Usefulness of SAT and Act Tests in Minority Admissions." Online Submission, Institute of Educational Sciences, 31 Dec. 2009, eric.ed.gov/?id=ED510111.

Thomas, M.K. (2004), The SAT II: Minority/Majority Test-Score Gaps and What They Could Mean for College Admissions. Social Science Quarterly, 85: 1318-1334. https://doi.org/10.1111/j.0038-4941.2004.00278.x