Capstone Project

Daniel Albert

3/20/2019

Introduction

In thinking about this project, I searched Kaggle for a promising dataset and found one entitled Student Performance in Exams by SPScientist (https://www.kaggle.com/spscientist/students-performance-in-exams). This dataset includes anonymous information on 1000 students including their gender, race category (the categories are not specific), highest level of parental education, whether or not they qualify for free/reduced lunches, whether or not they completed a preparatory course, and their scores on three exams: mathematics, reading, and writing. I wondered if we could use the demographic information to predict their performance on any/all of the exams. As a teacher, I know that there is a lot of discussion about "closing the gap" in performance between various subsets of students. While all teachers know of exceptions to these generalities, this is something that schools do look into and I was wondering if I could find a useful algorithm.

Preliminary Analysis

After downloading the data, I split it into test and training sets, then further split the training set so that I could play around with things without testing against the sacred test set. I named my practice set "data" and my practice test set "further_test_set".

```
data_path <- "~/Capstone/StudentsPerformance.csv"
data_original <- read.csv(data_path)
set.seed(24)
test_index <- createDataPartition(y = data_original$math.score, times = 1, p = .2, list = FALSE)
train_set <- data_original[-test_index,]
test_set <- data_original[test_index,]
further_test_index <- createDataPartition(train_set$math.score, times = 1, p = .2, list = FALSE)
further_test_set <- train_set[further_test_index,]
data <- train_set[-further_test_index,]
```

Now that we have that set up, I did some exploring of data. I found the basic averages for each subject and stored this information.

```
mu_math <- mean(data$math.score)
mu_reading <- mean(data$reading.score)
mu_writing <- mean(data$writing.score)
averages <- c(mu_math, mu_reading, mu_writing)
averages
```

```
## [1] 66.09591 69.11478 67.93553
```

We can see that the average score on the math exam was lower than the other two. I plan on using these averages in calculations ahead, so I want a data frame with those values on each line.

```
averages <- t(data.frame(averages, averages, averages, averages, averages, averages))
```

At this point I began looking at the different predictors we have access to. I separated them one at a time to see if any had a particularly strong impact on the scores. In each case, the delta values I looked at in the end are the deviations from the average. A positive number indicates higher than average.

```
gender_averages <- data %>% group_by(gender) %>% summarize(math.gender = mean(math.score), reading.gende
gender_labels <- gender_averages[,1]
gender_delta <- gender_averages[,2:4] - averages[1:2,]
gender_delta <- bind_cols(gender_labels, gender_delta)
gender_delta
```

```
## # A tibble: 2 x 4
##   gender math.gender reading.gender writing.gender
##   <fct>        <dbl>          <dbl>          <dbl>
## 1 female       -2.07           3.75           4.87
## 2 male          2.28          -4.15          -5.39
```

Here we see that female students are below average in math but above in reading and writing while male students are the reverse.

```
race_averages <- data %>% group_by(race.ethnicity) %>% summarize(math.race = mean(math.score), reading.
race_labels <- race_averages[,1]
race_delta <- race_averages[,2:4] - averages[1:5,]
race_delta <- bind_cols(race_labels, race_delta)
race_delta
```

```
## # A tibble: 5 x 4
##   race.ethnicity math.race reading.race writing.race
##   <fct>              <dbl>        <dbl>        <dbl>
## 1 group A            -5.70       -5.88        -6.83
## 2 group B            -3.69       -2.64        -3.82
## 3 group C            -1.72        0.473        0.231
## 4 group D             1.88        1.10         2.46
## 5 group E             8.40        3.85         4.10
```

Looking at the race information, we can see that groups A and B are below average in everything while D and E are above. One might guess which races are which at this point, but I will refrain from making assumptions based on stereotypes.

```
parent_averages <- data %>% group_by(parental.level.of.education) %>% summarize(math.parent = mean(math
parent_labels <- parent_averages[,1]
parent_delta <- parent_averages[,2:4] - averages[1:6,]
parent_delta <- bind_cols(parent_labels, parent_delta)
parent_delta
```

```
## # A tibble: 6 x 4
##   parental.level.of.education math.parent reading.parent writing.parent
##   <fct>                             <dbl>          <dbl>          <dbl>
## 1 associate's degree                0.937          1.12           1.04
## 2 bachelor's degree                 3.88           4.69           6.35
## 3 high school                      -3.87          -4.17          -5.65
## 4 master's degree                   4.98           6.58           8.19
## 5 some college                      1.30           0.255          0.805
## 6 some high school                 -4.18          -3.78          -4.84
```

Although this is ordered alphabetically rather than in terms of increasing education, we see that the expected is true. On average, parents who are more educated have higher performing children.

```
lunch_averages <- data %>% group_by(lunch) %>% summarize(math.lunch = mean(math.score), reading.lunch =
lunch_labels <-lunch_averages[,1]
lunch_delta <- lunch_averages[,2:4] - averages[1:2,]
lunch_delta <- bind_cols(lunch_labels, lunch_delta)
```

```
lunch_delta
```

```
## # A tibble: 2 x 4
##   lunch        math.lunch reading.lunch writing.lunch
##   <fct>             <dbl>         <dbl>         <dbl>
## 1 free/reduced      -7.16         -3.93         -4.33
## 2 standard           3.84          2.11          2.32
```

Here we see that there is a rather large discrepency between those who are on free/reduced lunch and those who are not. This may be why this is a demographic of particular interest to schools.

```
prep_averages <- data %>% group_by(test.preparation.course) %>% summarize(math.prep = mean(math.score),
prep_labels <-prep_averages[,1]
prep_delta <- prep_averages[,2:4] - averages[1:2,]
prep_delta <- bind_cols(prep_labels, prep_delta)
prep_delta
```

```
## # A tibble: 2 x 4
##   test.preparation.course math.prep reading.prep writing.prep
##   <fct>                       <dbl>        <dbl>        <dbl>
## 1 completed                    3.08         4.60         6.39
## 2 none                        -1.70        -2.53        -3.52
```

Once again we see the expected results. Those students who completed a prep course performed better than those who did not. This is most noticeable in the writing category.

I'm going to the RMSE metric to decide which method is best. I had explord using another method, the percentage of predictions within 5, 10, and 20 points of the true value, but this value did not significantly change in my tests, so I will omit it.

```
RMSE <- function(x, y){sqrt(mean((x-y)^2))}
```

Methods

Method 0

As a baseline, I'm going to call Method 0 "Just the Averages" and predict the math average for all math exams, reading average for all reading exams, etc.

```
results <- data.frame(method = "Just the Average", math_rmse = RMSE(mu_math, further_test_set$math.score

results
```

```
##             method math_rmse reading_rmse writing_rmse
## 1 Just the Average  15.92606     15.15118      15.4559
```

So here we see that the RMSE of this method is between 15 and 16 for the different subjects. My goal is to improve this in other methods.

Method 1: Factor Averages

Seeing as how I have calculated the average impact of each factor on scores, I thought we might sum these for each student in order to predict their grades. To do this, I will add columns to the further_test_set for each of the predictors. I will then sum these along with the average for the appropriate subject to make my prediction.

```
further_test_set <- further_test_set %>% left_join(gender_delta, by = 'gender') %>%
  left_join(race_delta, by = 'race.ethnicity') %>%
  left_join(parent_delta, by= 'parental.level.of.education') %>%
  left_join(lunch_delta, by = 'lunch') %>%
  left_join(prep_delta, by = 'test.preparation.course')
math_pred <- further_test_set %>%  mutate(pred = mu_math + math.gender + math.parent + math.race + math
reading_pred <- further_test_set %>% mutate(pred = mu_reading + reading.gender + reading.race + reading
writing_pred <- further_test_set %>% mutate(pred = mu_writing + writing.gender + writing.race + writing
results <- bind_rows(results, data.frame(method = "Factor Averages", math_rmse = RMSE(math_pred, furthe
results
```

```
##             method math_rmse reading_rmse writing_rmse
## 1 Just the Average  15.92606     15.15118     15.45590
## 2  Factor Averages  13.62677     13.92264     13.43369
```

We see a distinct improvement here, getting the RMSEs down by a couple of points.

Method 2: Regression Tree

Another method to try is a Regression Tree. For this one, I wanted to separate out the different subjects so that they wouldn't be confused for predictors. I ran each one individually through rpart and then used them to make predictions and found the associated RMSEs.

```
just_math <- select(data, -c(reading.score, writing.score))
just_reading <- select(data, -c(math.score, writing.score))
just_writing <- select(data, -c(math.score, reading.score))

fit_math <- rpart(math.score ~., data = just_math)
tree_math <- predict(fit_math, further_test_set)

fit_reading <- rpart(reading.score ~., data = just_reading)
tree_reading <- predict(fit_reading, further_test_set)

fit_writing <- rpart(writing.score ~., data = just_writing)
tree_writing <- predict(fit_writing, further_test_set)

tree_rmse <- c(RMSE(tree_math, further_test_set$math.score), RMSE(tree_reading, further_test_set$reading
results <- bind_rows(results, data.frame(method = "Tree", math_rmse = tree_rmse[1],
                                         reading_rmse = tree_rmse[2],
                                         writing_rmse = tree_rmse[3]))
results[2:3,]
```

```
##            method math_rmse reading_rmse writing_rmse
## 2 Factor Averages  13.62677     13.92264     13.43369
## 3            Tree  14.51654     14.22988     13.65008
```

Here we see that our predictions for math and reading are worse across the board, though reading and writing are not far off. Perhaps a random forest would improve upon this?

Method 3: Random Forest

```
rf_fit_math <- randomForest(math.score ~ ., data = just_math)
rf_pred_math <- predict(rf_fit_math, further_test_set)
rf_fit_reading <- randomForest(reading.score ~., data = just_reading)
```

```
rf_pred_reading <- predict(rf_fit_reading, further_test_set)
rf_fit_writing <- randomForest(writing.score ~., data = just_writing)
rf_pred_writing <- predict(rf_fit_writing, further_test_set)
#
results <- bind_rows(results, data.frame(method = "Random Forest",
                                         math_rmse = RMSE(rf_pred_math, further_test_set$math.score),
                                         reading_rmse = RMSE(rf_pred_reading, further_test_set$reading.s
                                         writing_rmse = RMSE(rf_pred_writing, further_test_set$writing.s
results[2:4,]
```

```
##            method math_rmse reading_rmse writing_rmse
## 2 Factor Averages  13.62677     13.92264     13.43369
## 3            Tree  14.51654     14.22988     13.65008
## 4   Random Forest  14.15845     13.91194     13.58077
```

Random Forests are an improvement over Regression Trees, but it's still slightly behind my Factor Averages method in math and writing and practically the same in reading.

Results

In the end, I have chosen to try the Factor Averages with the real test set.

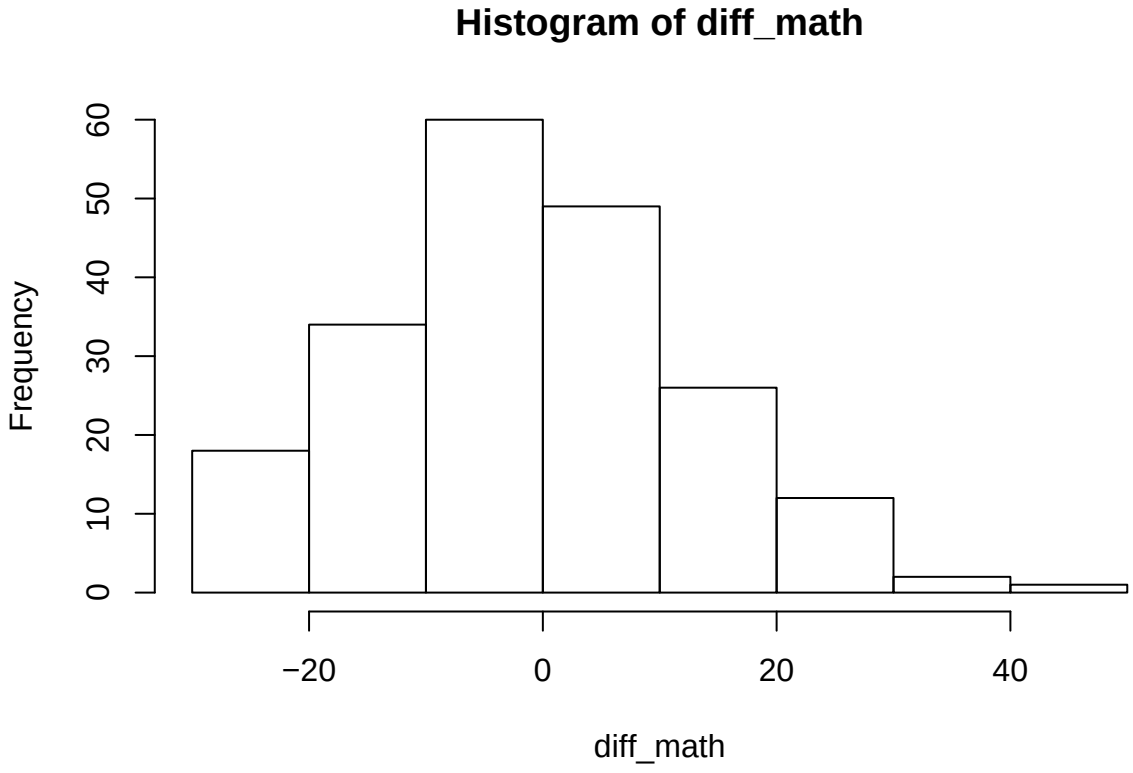
```
mu_math2 <- mean(train_set$math.score)
mu_reading2 <- mean(train_set$reading.score)
mu_writing2 <- mean(train_set$writing.score)

averages2 <- c(mu_math2, mu_reading2, mu_writing2)
#I'm going to use the averages data frame ahead, so I need each row to be those three averages
averages2 <- t(data.frame(averages2, averages2, averages2, averages2, averages2, averages2))

#Computing the factor averages for use, as above
gender_averages2 <- train_set %>% group_by(gender) %>% summarize(math.gender = mean(math.score), reading
gender_delta2 <- gender_averages2[,2:4] - averages[1:2,]
gender_delta2 <- bind_cols(gender_labels, gender_delta2)

race_averages2 <- train_set %>% group_by(race.ethnicity) %>% summarize(math.race = mean(math.score), rea
race_delta2 <- race_averages2[,2:4] - averages[1:5,]
race_delta2 <- bind_cols(race_labels, race_delta2)


parent_averages2 <- train_set %>% group_by(parental.level.of.education) %>% summarize(math.parent = mean
parent_delta2 <- parent_averages2[,2:4] - averages[1:6,]
parent_delta2 <- bind_cols(parent_labels, parent_delta2)


lunch_averages2 <- train_set %>% group_by(lunch) %>% summarize(math.lunch = mean(math.score), reading.lu
lunch_delta2 <- lunch_averages2[,2:4] - averages[1:2,]
lunch_delta2 <- bind_cols(lunch_labels, lunch_delta2)


prep_averages2 <- train_set %>% group_by(test.preparation.course) %>% summarize(math.prep = mean(math.s
prep_delta2 <- prep_averages2[,2:4] - averages[1:2,]
prep_delta2 <- bind_cols(prep_labels, prep_delta2)
```

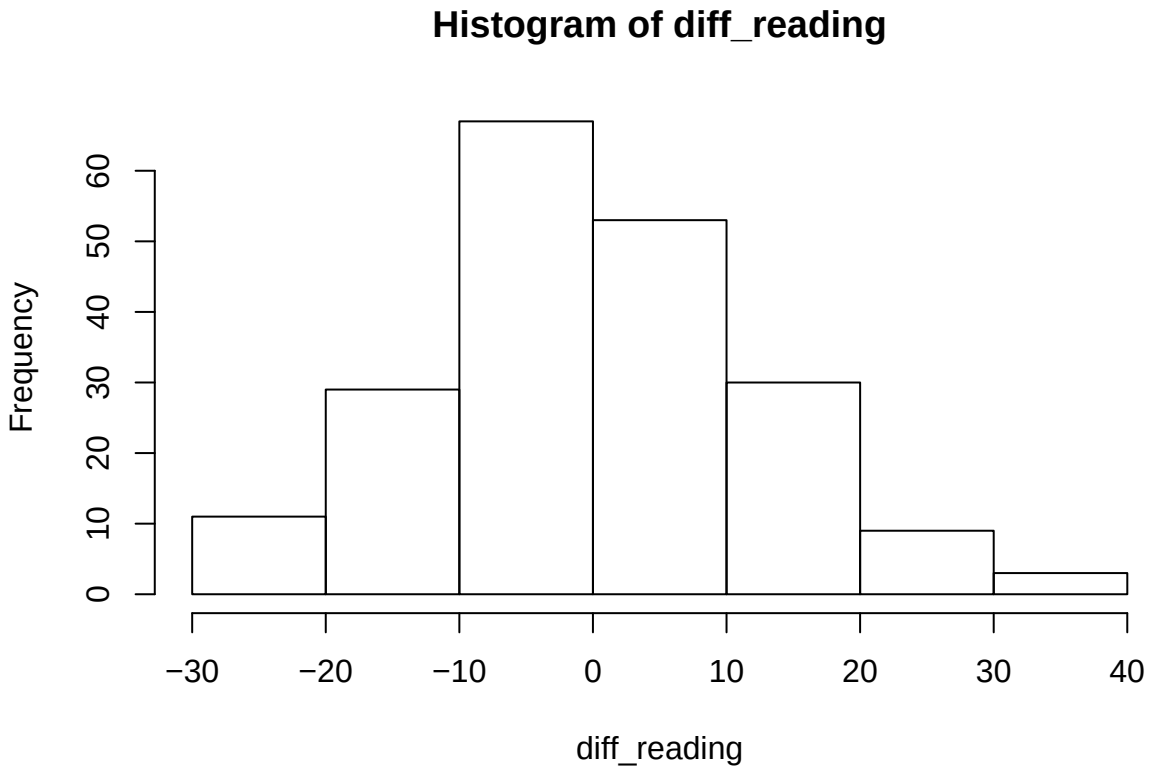
```r
#add these columns onto our test_set and make our predictions!
test_set <- test_set %>% left_join(gender_delta2, by = 'gender') %>%
  left_join(race_delta2, by = 'race.ethnicity') %>%
  left_join(parent_delta2, by= 'parental.level.of.education') %>%
  left_join(lunch_delta2, by = 'lunch') %>%
  left_join(prep_delta2, by = 'test.preparation.course')

#now I can calculate my predictions based on the averages of the factors
math_pred2 <- test_set %>%  mutate(pred = mu_math2 + math.gender + math.parent + math.race + math.lunch
reading_pred2 <- test_set %>% mutate(pred = mu_reading2 + reading.gender + reading.race + reading.paren
writing_pred2 <- test_set %>% mutate(pred = mu_writing2 + writing.gender + writing.race + writing.paren


final_results <- data.frame(method = "Factor Averages",
                            math_rmse = RMSE(math_pred2, test_set$math.score),
                            reading_rmse = RMSE(reading_pred2, test_set$reading.score),
                            writing_rmse = RMSE(writing_pred2, test_set$writing.score))

final_results
```

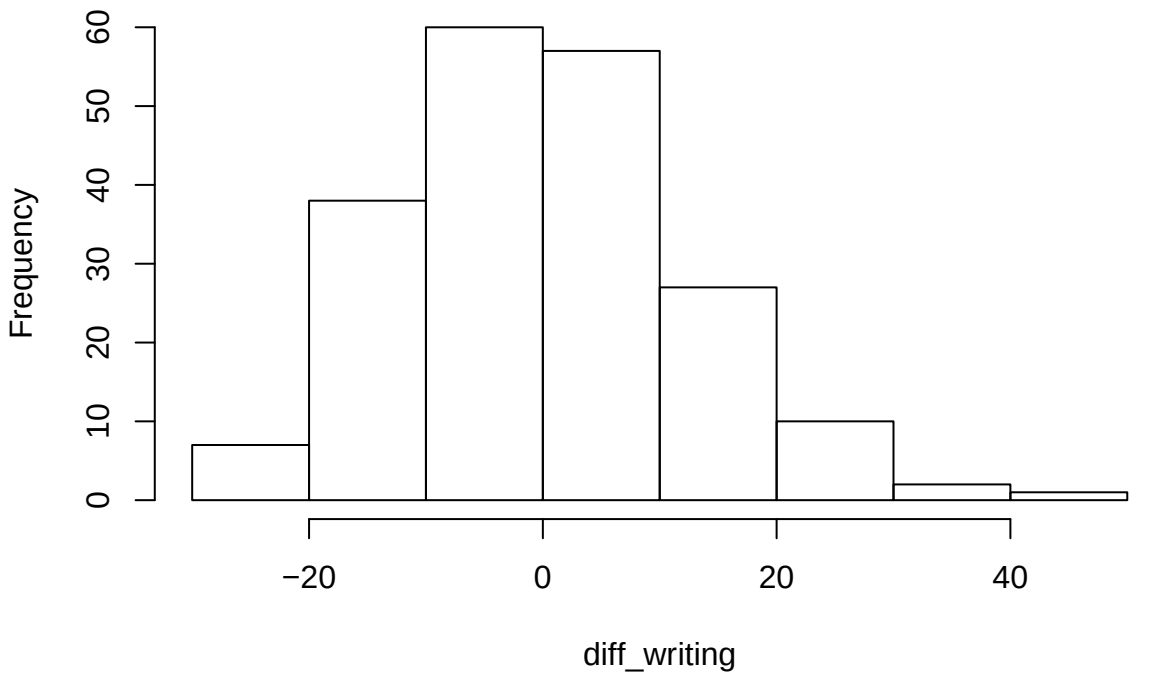
```
##            method math_rmse reading_rmse writing_rmse
## 1 Factor Averages  13.57434     12.33391     12.56237
```

I can now look a little deeper into the data and examine the other metric. I will calculate how far off each prediction was on each subject. Here is a graph of each subject's differential. In these instance, positive numbers means that we guessed too high (the student scored lower than we predicted) and vice versa.

```r
diff_math <- math_pred2 - test_set$math.score
diff_reading <- reading_pred2 - test_set$reading.score
diff_writing <- writing_pred2 - test_set$writing.score
```







Each of these shows the distribution of how far off our predictions were. We can see that our estimates peak slightly below the actual value, meaning that the largest chunk of students score slightly better than we predicted. That being said, none of the graphs show distributions that are very far off center. These graphs are useful visuals, but I also want to look at the decimal representations of how close we were.

```
diff_summary <- data.frame(subject = "math", "within 5" = mean(abs(diff_math) <=5), "within 10" = mean(a
diff_summary <- bind_rows(diff_summary, data.frame(subject = "reading", "within 5" = mean(abs(diff_read
diff_summary <- bind_rows(diff_summary, data.frame(subject = "writing", "within 5" = mean(abs(diff_writ
diff_summary
```

```
##   subject  within.5 within.10 within.20
## 1    math 0.2425743 0.5396040 0.8366337
## 2 reading 0.3613861 0.5940594 0.8861386
## 3 writing 0.3019802 0.5792079 0.9009901
```

Here we can see that our predictions only fell within 10 points of the actual value between 50% and 60% of the time. This means that if we predicted that a student was going to earn a C or a D on an exam, there was only a 50-60% chance that we were right. That's really not very good. As I said earlier, I did not include this metric in the rest of this report because the numbers changed very little as I tried different methods.

Conclusion

These are the best RMSE numbers we've seen so far, but they're still larger than I'd like. I think that there are a few things to consider when looking at the results:

First, it is a relatively small data set, only 1000 students. If we had significanatly more data, I would have tried separating things so that we could calculate an average for each combination of factors (for example, we could calculate the average of female students of race C whose parents have a bachelor's degree, who does not qualify for free lunch, and who has not completed a prep class), but I felt that there was insufficient data for this approach.

Second, it may well be that there is no "good" method for predicting student success from these factors. The education world is well aware of many of the trends discovered here, but as any teacher will tell you, there are exceptions to these trends, sometimes more so than the norm. While closing acheivement gaps is an important task, targeting students based solely on their race or parents' education is going to unnecessarily catch students who don't require intervention as well as miss other students who do. I think that a better way to target students for intervention is to ask the student's teachers and to get to know the students individually (which, admittedly, takes longer than looking at their demographic information) and to give them appropriate help. It isn't the answer that people want, but it's the answer that we have.