The College Scorecard

Varun Nadgir

September 27, 2017

Capstone Project

Introduction

The **College Scorecard** is a service meant to help prospective students make their college decision. Whether by comparing size, popular majors, or comparing costs to the national average, the site's goal is to help the user find a good fit. For my Springboard Capstone Project, I used the **dataset** made available by the College Scorecard to try and find additional ways to help users in their decision.

The Data

Available in a .zip file from the link above, the data is split into 19 .csv files - one for each academic year from 1996-'97 to 2014-'15. Each file contains 1,744 recorded data points (columns) and about 7,500 schools (rows). My first step was to add a DATAYEAR column to indicate the academic year and then merging the 19 files into one large .csv, which I called **fulldata.csv**. From this 2 GB file, I would create subsets for plotting and studying trends. Next, I had to refer to **the data dictionary** to understand the column names and some of the placeholder values used. Once I merged the files and had a basic understanding of what data was available, I began to create some plots and documented my initial findings in my **data story**. As a note, my studies have been on US schools only. The dataset includes records of US territories as well, but I have filtered them out when creating subsets for plotting/modeling.

Deliverables

My first item will be using linear regression models to determine what variables are the most influential on the cost of a school, and to potentially predict what the cost of a school may be in the future. This would be useful in two ways. It could help students who are on the fence about going to college immediately after high school by suggesting a decline in cost. If a student sees that their ideal school is likely to be cheaper in two years, they may make the decision to find entry-level work or go traveling before going to college. It could also help the schools by indicating areas of their budget that are influencing the cost of attendance. Of course, the goals of each school are different and they may not be interested in reducing cost, but if a school is experiencing a decline in applications, cost of attendance is likely to be something they look at.

The second item is a recommendation tool that works similarly to clustering methods used by media services such as Netflix, YouTube, and iTunes. Clustering based on things like location, cost, and SAT scores, a student can find options that are close to their top choice. Mentality is a very important part of finding success at school, and feeling out of place in freshman year can be quite discouraging. If their top choice is a far reach school, or it is too expensive/too far, then finding alternatives would hopefully help them to be satisfied in their decision.

My final item will be a basic UI that allows the user to explore the dataset on their own. Although the data will need to be curated and shaved down to a size that a standard internet browser can handle, my hope is that it will provide some transparency between students and universities. As an example, in my **data story**, I explored SAT averages and admission rates. Even though one could reasonably guess how they are related (higher SAT scores ~ lower admission rates), being able to plot the data and draw a conclusion from a graph is much more convincing. By putting this power in the hands of students and their families, they should be able to make much more educated decisions.

Data Preparation

From **fulldata.csv**, I created and saved a subset of the columns that would make repeated data manipulation more manageable. I named it **subdata.csv** and it contains various *school identification data* (name, ID, location, level of institution), *admission data* (student demographics, SAT scores), *education data* (majors, completion rates), and *financial data* (cost, aid, debt, repayment).

From this subset, I can pick the variables for the cost modeling, as well as the variables for clustering - both tools will get specific subsets made so that the data is organized and loading it will be easier in the future. A combination of the original source data and my curated data will go into the UI plotting tool, keeping in mind that the dataset should stay a reasonable size but also have enough depth for the user to gain whatever insights they can.

Cost Prediction

For convenience, I began by changing the DATAYEAR column from indicating the year range ("1996-'97") to simply having the start year ("1996"). Using the datayear, I created the training file using years 1996 to 2013 and the test file with just the 2014 data. Since the cost is known for 2014, we can compare afterwards to see how good the model is beyond the R^2 score alone.

From **subdata.csv**, I created a subset of up to 14 columns that I felt would be related to the attendance cost of a school. This included columns like "CONTROL" (whether the school is public or private), "UGDS" (the undergraduate student population), "PFTFAC" (percentage of faculty that are full time), and of course "COSTT4_A" (cost of attendance for a 4-year academic year school). Upon checking the summary() of the training set, I noticed that the MAX of "UGDS" was unnaturally high. This was because there were a few records of the University of Pheonix's online campus, which is capable of having about 200,000 students. Since this was skewing the data, I removed its related "OPEID6" from the training and test sets. I also removed all records in the training set that did not have a "COSTT4_A" entry, since they would not contribute to the model. This filtering had an interesting byproduct - apparently years 2008 and prior did not have "COSTT4_A" records, so removing the NA entries also happened to filter the dataset to years 2009 and beyond. After taking care to turn appropriate columns into the numeric data type, I moved on to make small experimental models.

The first model I tried was on Cost by Undergrad Pop., Average Faculty Salary, and % of Faculty that are Full-Time. This, however, only gave an R^2 of 0.2198, which is not that great. However, the model summary suggests that these three variables are still significant (using asterisk notation), so it is a step in the right direction. The 2nd model I tried used 12 independent variables to model Cost. It gave an R^2 of 0.8253, but still showed that three of the variables had little to no significance. After removing these, this was the final model I came to:

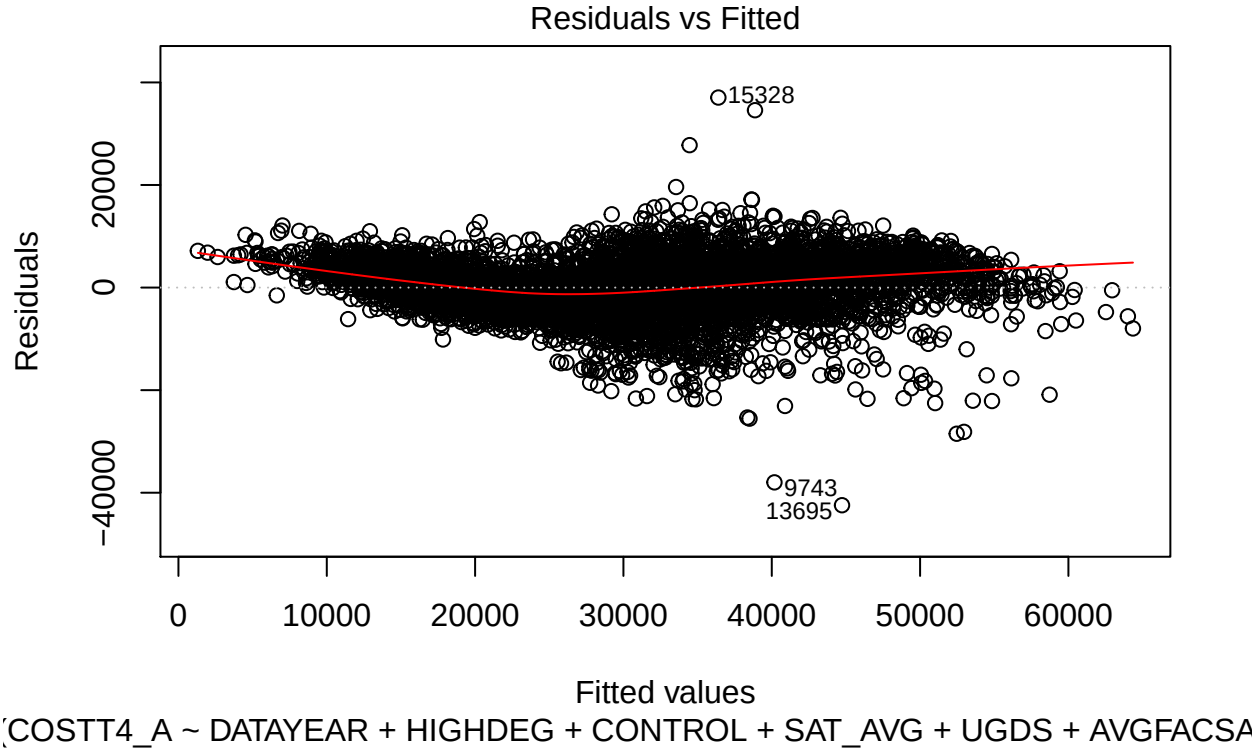
Cost by Datayear, Highest degree offered, Pub/Priv, Avg SAT, Undergrad Pop., Avg Faculty Salary, % of Faculty that are Full-Time, Median Debt, and Avg Family Income

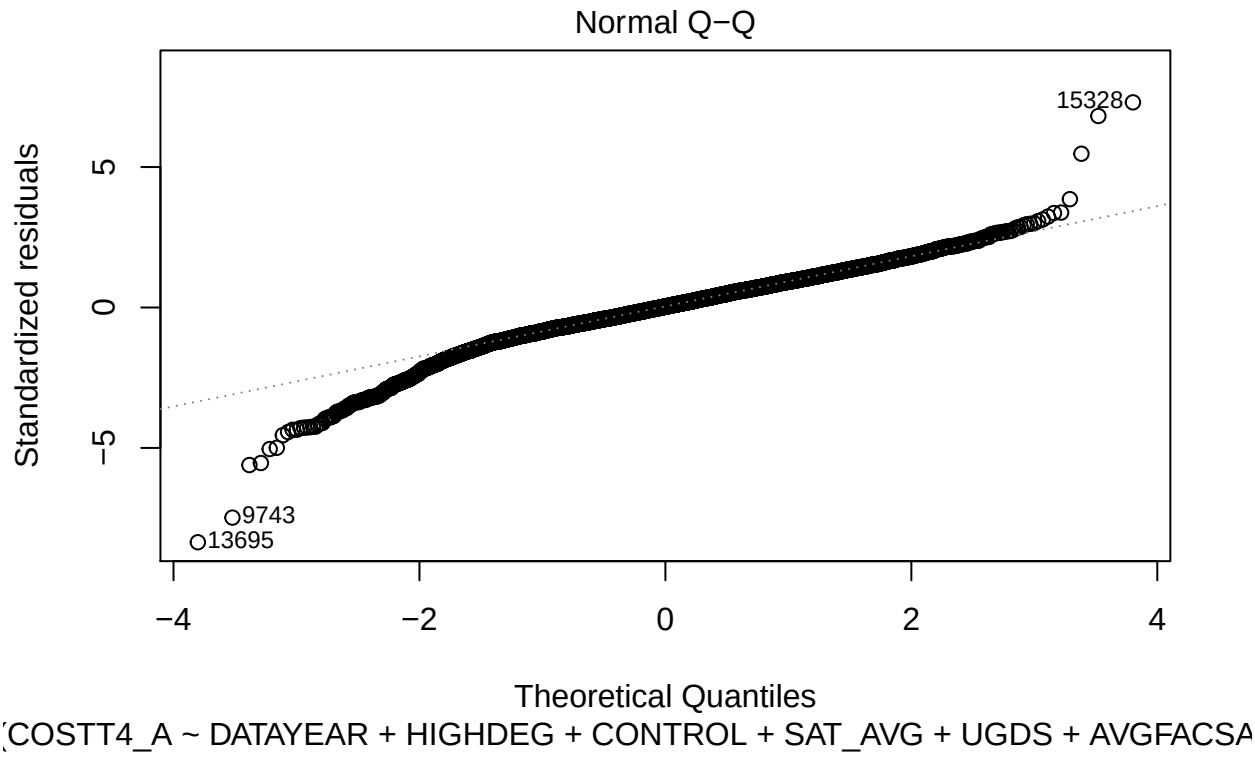
```
##
## Call:
## lm(formula = COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + SAT_AVG +
##     UGDS + AVGFACSAL + PFTFAC + DEBT_MDN + FAMINC, data = df.train,
##     na.action = na.omit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -42436  -2843    163   3267  37049
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -1.510e+06  1.013e+05 -14.914  < 2e-16 ***
## DATAYEAR     7.357e+02  5.040e+01  14.597  < 2e-16 ***
## HIGHDEG      7.880e+02  1.277e+02   6.171 7.19e-10 ***
## CONTROL      1.522e+04  1.606e+02  94.793  < 2e-16 ***
## SAT_AVG      1.140e+01  8.111e-01  14.059  < 2e-16 ***
## UGDS        -1.843e-01  1.155e-02 -15.954  < 2e-16 ***
## AVGFACSAL    1.709e+00  4.681e-02  36.509  < 2e-16 ***
## PFTFAC      -1.352e+03  2.673e+02  -5.056 4.39e-07 ***
## DEBT_MDN     2.558e-01  1.992e-02  12.838  < 2e-16 ***
## FAMINC       1.043e-01  4.618e-03  22.588  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5086 on 6962 degrees of freedom
##   (12666 observations deleted due to missingness)
## Multiple R-squared:  0.825,  Adjusted R-squared:  0.8247
## F-statistic:  3646 on 9 and 6962 DF,  p-value: < 2.2e-16
```

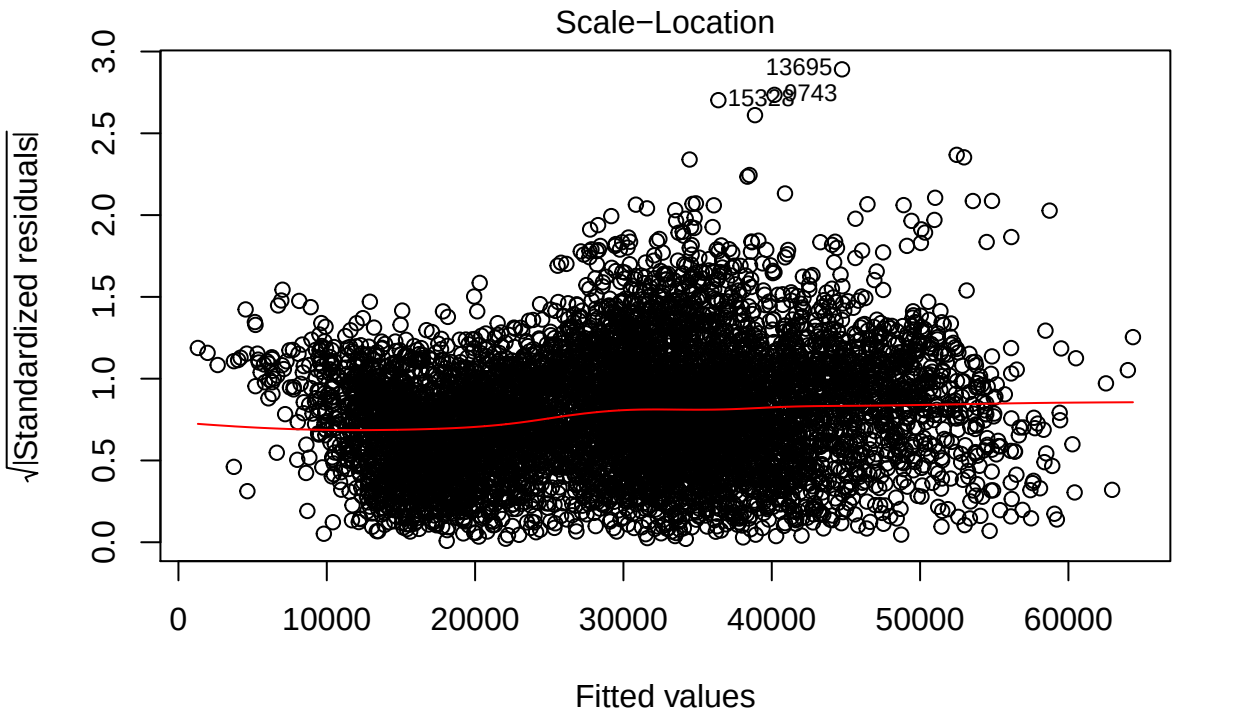
With an R^2 of 0.825, this model is almost equally as good as the second model, but now all of the variables used are significant. In other words, the variables that influence the cost of attendance of a school are the ones in this model. This is saved as **cost.mod** and will be used to predict on the test set. Before predicting, however, we should take a look at the plot of this model so that we get some visual context on how good or bad this fit is.

Plot of cost.mod

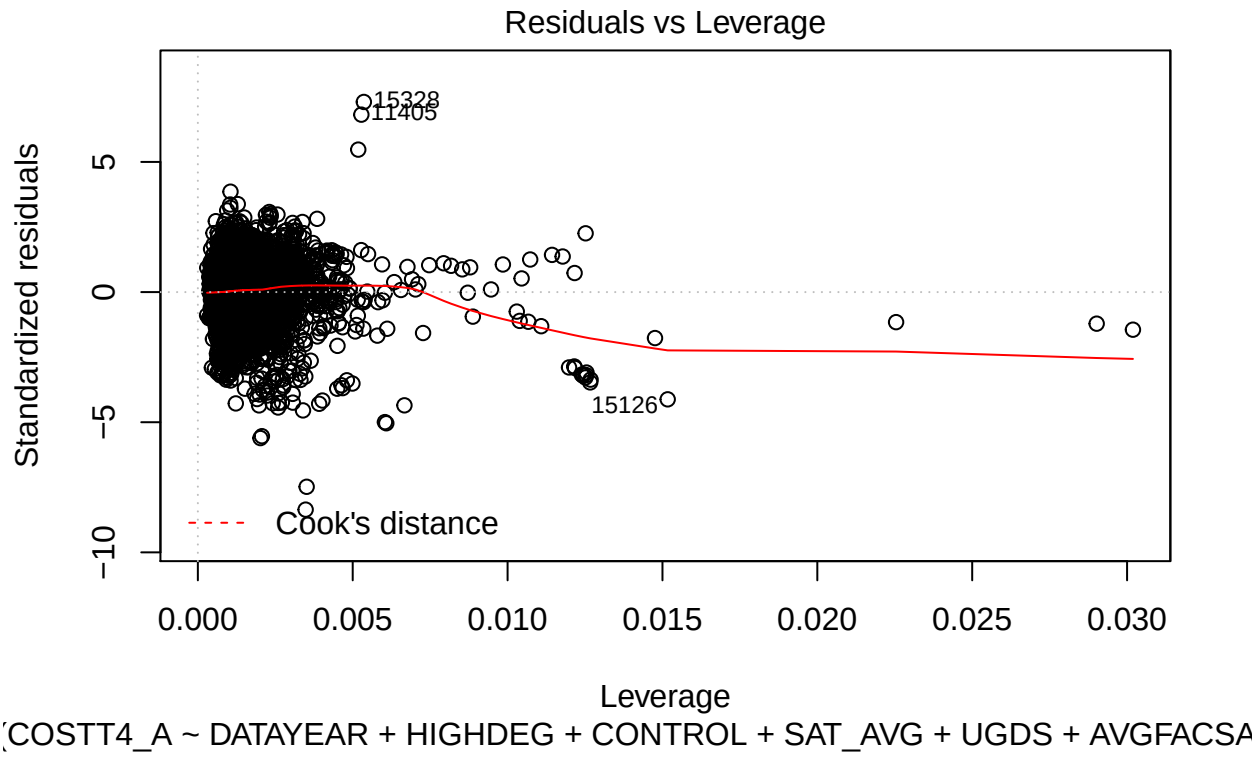




(COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + SAT_AVG + UGDS + AVGFACSA



(COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + SAT_AVG + UGDS + AVGFACSA



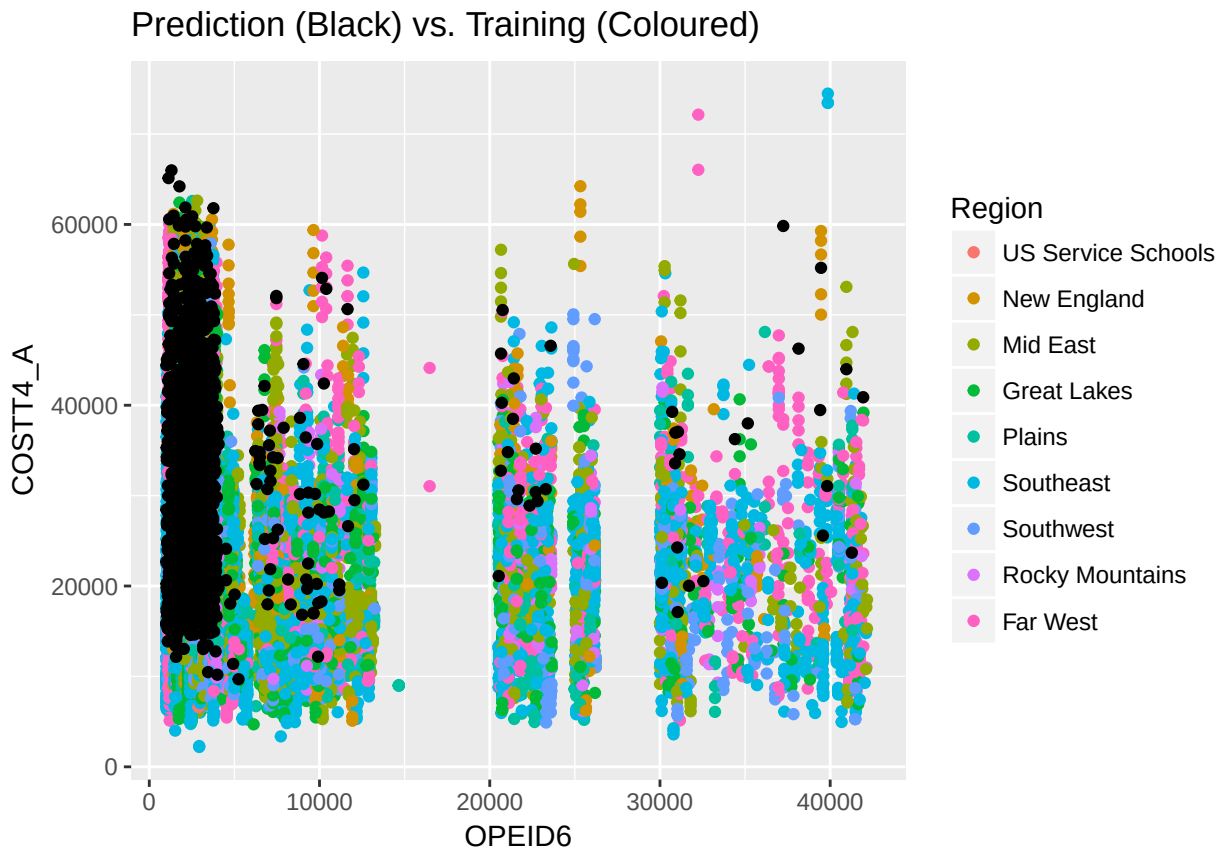
The Residuals vs Fitted plot stands out right away, showing that the residuals generally reside around the fitted line. The Normal Q-Q plot also follows the normal line for the most part, aside from the very bottom and very top of the plot. It seems like this model is fairly strong, and the R^2 and plot of residuals reinforces it. Now we can make some predictions on the 2014 data and make some comparisons.

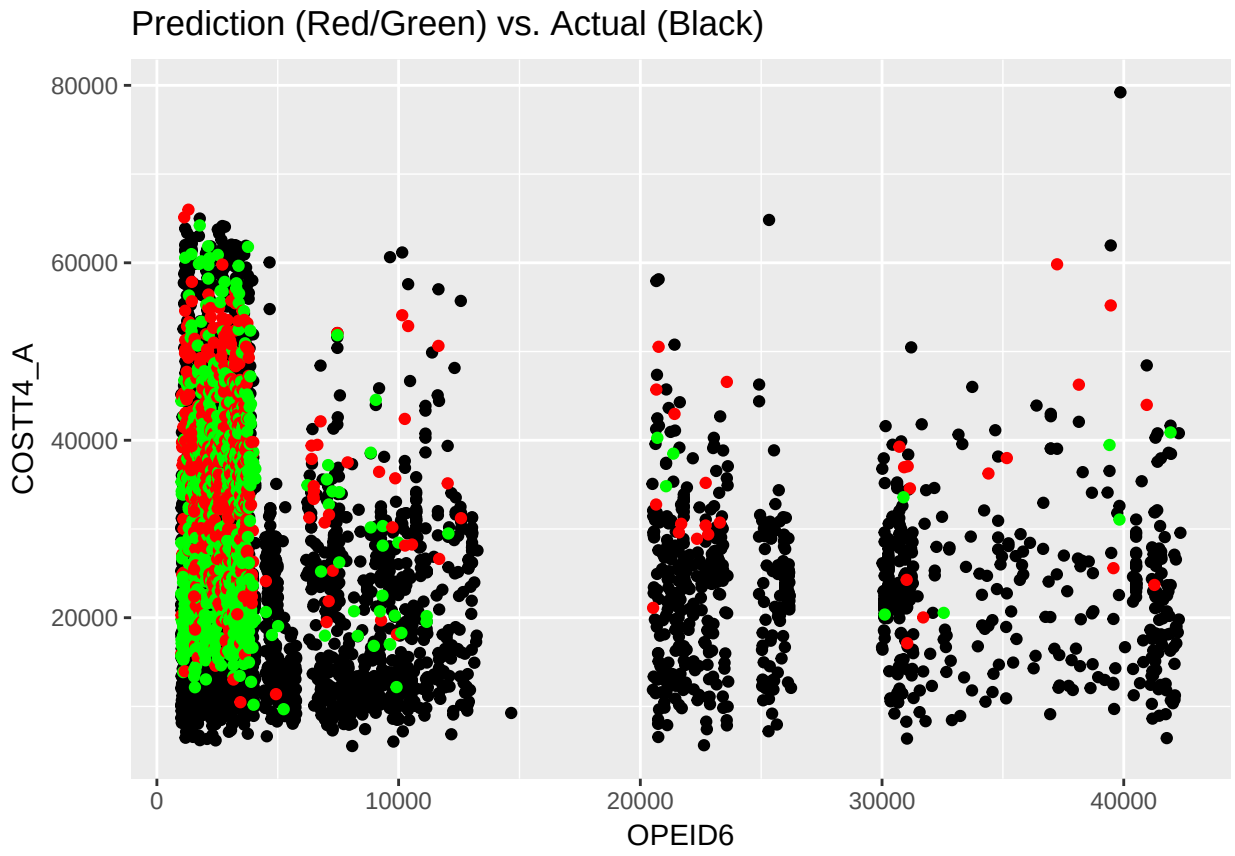
Summary of test data after predictions

```
##      OPEID6          INSTNM             DATAYEAR        REGION
##  Min.   : 1002   Length:3888        Min.   :2014   Min.   :0.000
##  1st Qu.: 2470   Class :character   1st Qu.:2014   1st Qu.:3.000
##  Median : 3810   Mode  :character   Median :2014   Median :5.000
##  Mean   : 9543                      Mean   :2014   Mean   :4.408
##  3rd Qu.:10727                      3rd Qu.:2014   3rd Qu.:6.000
##  Max.   :42345                      Max.   :2014   Max.   :8.000
##
##     HIGHDEG         CONTROL         SAT_AVG          UGDS
##  Min.   :0.00   Min.   :1.000   Min.   : 720   Min.   :    0
##  1st Qu.:2.00   1st Qu.:1.000   1st Qu.: 973   1st Qu.:  446
##  Median :3.00   Median :2.000   Median :1039   Median : 1486
##  Mean   :2.97   Mean   :1.833   Mean   :1059   Mean   : 3893
##  3rd Qu.:4.00   3rd Qu.:2.000   3rd Qu.:1120   3rd Qu.: 4294
##  Max.   :4.00   Max.   :3.000   Max.   :1545   Max.   :77657
##                                 NA's   :2595   NA's   :1
##    AVGFACSAL         PFTFAC          DEBT_MDN         FAMINC
##  Min.   :  332   Min.   :0.0000   Min.   : 1354   Min.   :     0
##  1st Qu.: 4914   1st Qu.:0.2958   1st Qu.: 8414   1st Qu.: 24289
```

```
##  Median : 6102   Median :0.5296   Median :12500   Median : 35595
##  Mean   : 6351   Mean   :0.5630   Mean   :12954   Mean   : 43701
##  3rd Qu.: 7553   3rd Qu.:0.8564   3rd Qu.:17500   3rd Qu.: 58456
##  Max.   :20650   Max.   :1.0000   Max.   :37500   Max.   :152100
##  NA's   :190     NA's   :520      NA's   :232     NA's   :15
##     COSTT4_A        PREDICT           DIFF
##  Min.   : 5536   Min.   : 9691   Min.   :-15605.9
##  1st Qu.:14634   1st Qu.:24010   1st Qu.: -2884.8
##  Median :23200   Median :34152   Median :   568.9
##  Mean   :25242   Mean   :33199   Mean   :   560.2
##  3rd Qu.:31789   3rd Qu.:40676   3rd Qu.:  3819.4
##  Max.   :79212   Max.   :65985   Max.   : 26866.1
##                  NA's   :2608    NA's   :2608
```

The predict() formula returns a column that I named "PREDICT" and, by taking the difference of "PREDICT" and "COSTT4_A", we can see whether the prediction was too high or too low. This summary tells us that the extreme cases were -$15,605 and $26,866 off the mark. What is more disappointing, however, is the number of NAs in the prediction output. To see where the model is falling short, I can check two plots. The first plot overlays the prediction values (in black) over the training data. This helps to see if the predicted values stayed in the same range and may indicate another reason why some predictions weren't made. The second plot overlays the prediction values over the actual 2014 values, and they are coloured green if within $4000 of the actual cost and red if predicted too far. If the plot is mostly green, that would be a good sign.





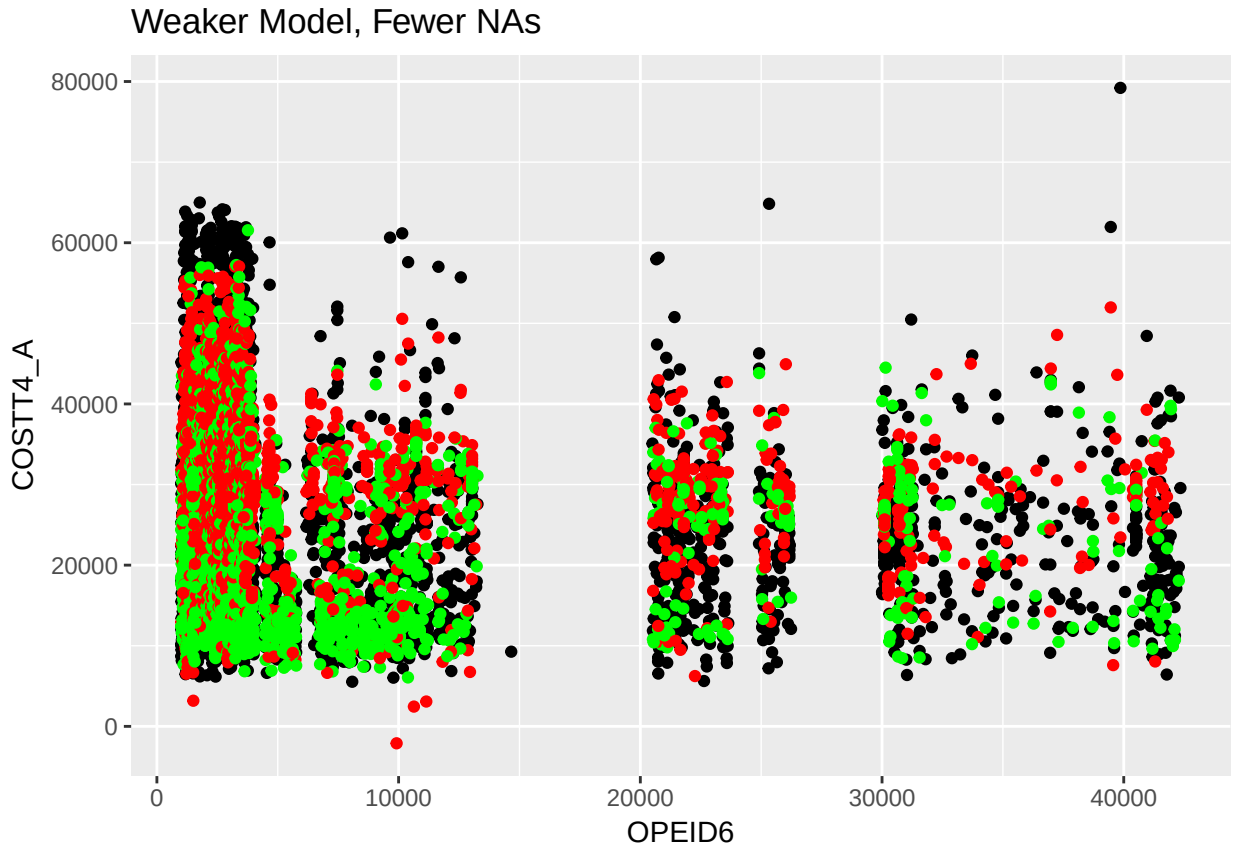
According to these plots, the predictions were mostly succesful for schools in the 0 ~ 3000 range for "OPEID6", while predictions beyond that are very sporadic. Looking at the dataset and sorting by "OPEID6" shows that there are many NAs in the independent variables, causing the prediction to fail for those cases. This is unfortunate, since this causes about 2,600 failed predictions despite a fairly strong model. This leaves us with a few different options. We could remove the independent variable(s) with the most NAs from the model, which risks weakening the model but will yield more successful predictions. Another option could be to replace NAs with another value, such as the mean for that column. This doesn't hurt the model at all, but the predictions for the schools with approximated values have a chance to be completely off.

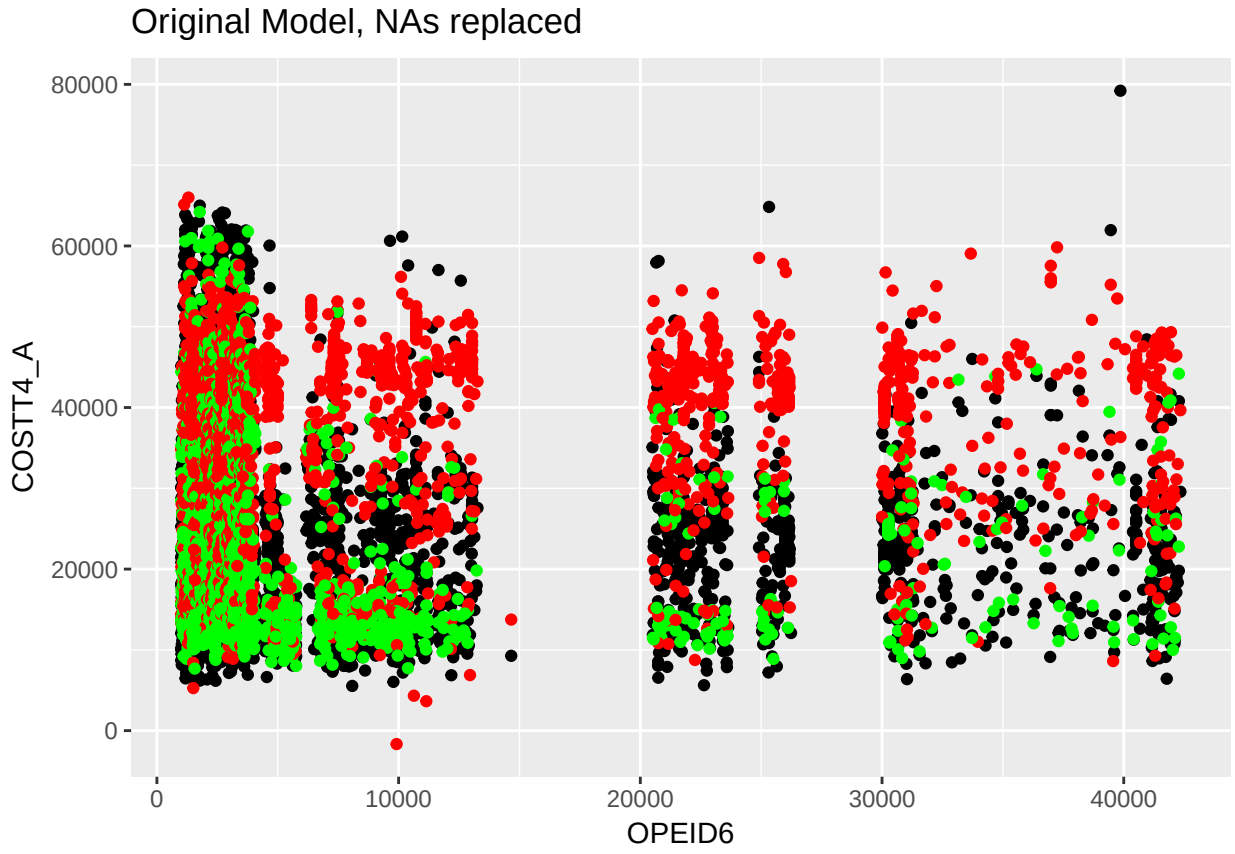
The independent variable with significantly more NAs than the others is "SAT_AVG". We can create a new model that does not include "SAT_AVG" and, it turns out, removing it reveals "PFTFAC" to be an insignificant variable as well. With both removed, the new model has an R^2 of 0.771, which is not bad at all. Checking the summary of this new model shows that there are only about 400 NAs in the prediction this time, which is a massive improvement.

```
##
## Call:
## lm(formula = COSTT4_A ~ DATAYEAR + HIGHDEG + CONTROL + UGDS +
##     AVGFACSAL + DEBT_MDN + FAMINC, data = df.train, na.action = na.omit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -47320  -3358    -87   3148  49616
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.144e+06  6.731e+04  -17.00   <2e-16 ***
```

```
## DATAYEAR     5.619e+02  3.348e+01   16.78   <2e-16 ***
## HIGHDEG      1.018e+03  5.861e+01   17.36   <2e-16 ***
## CONTROL      8.354e+03  7.065e+01  118.25   <2e-16 ***
## UGDS        -1.755e-01  8.043e-03  -21.82   <2e-16 ***
## AVGFACSAL    1.217e+00  3.119e-02   39.03   <2e-16 ***
## DEBT_MDN     2.816e-01  1.408e-02   19.99   <2e-16 ***
## FAMINC       2.134e-01  2.913e-03   73.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5660 on 17310 degrees of freedom
##   (2320 observations deleted due to missingness)
## Multiple R-squared:  0.771,  Adjusted R-squared:  0.7709
## F-statistic:  8327 on 7 and 17310 DF,  p-value: < 2.2e-16
```

We can compare the performance of this model to the other option, where NAs would have been replaced by the column mean. Is it worth it to have a slightly weaker model for more accurate predictions, or will replacing the NAs solve the problem? Ideally, the colour coding of the plot will indicate the better model.





School Recommendation

College Scorecard Sandbox