

# What distinguishes high-performing students?

*Will Landau*

*April 6, 2015*

## Introduction

The goal is to find and derive variables in the 2012 OECD PISA dataset that separate the highest-performing students from the lowest-performing ones in the United States.

In 2012, the Organization for Economic Co-operation and Development (OECD) Programme for International Student Assessment (PISA) surveyed roughly five hundred thousand, fifteen-year-old students from sixty-five economies across the globe (“Organization for Economic Co-operation and Development” 2015). Questions measured students’ reading, math, and science skills in ways that, according to the OECD website, “are not directly linked to the school curriculum. The tests are designed to assess to what extent students at the end of compulsory education, can apply their knowledge to real-life situations and be equipped for full participation in society” (“Organization for Economic Co-operation and Development” 2015). Students also answered extensive background questionnaires about their study habits, attitudes towards school, circumstances at home, etc. Extensive data were recorded about the schools and parents of those students as well.

PISA math and reading scores naturally divide USA subjects into high-performing students (top 25%) and low-performing students (bottom 25%). I will select a subset of variables in the student-specific PISA data and attempt to predict student performance (high or low) in the United States.

## Exploratory analysis

The student-specific USA data has roughly five hundred variables for predicting reading and math scores. After removing the few continuous variables and the variables with all missing values, I am left with 256 categorical variables for prediction. As seen in Figure 1, most variables still have a large fraction of missing values.

To explore the data further and 256 variables and build a classifier, I rank the variables according to a simple matching heuristic. The heuristic is similar to the simple matching coefficient between two categorical variables, and it loosely measures how well a factor accurately splits students according to high or low performance. For each factor  $x$ , I calculate this version of the matching coefficient as follows.

1. Remove the missing values from  $x$ , along with the corresponding values from the binary vector  $y$  of student performances (high and low coded as 1 and 0, respectively).
2. For every subset  $s$  of the levels of  $x = (x_1, \dots, x_n)$ :
  - a. Create the binary vector  $z = (z_1, \dots, z_n)$ , where  $z_i = I(x_i \in s)$ .
  - b. Let the matching score of  $s$  be

$$\frac{1}{n} \max \left\{ \sum_{i=1}^n I(y_i = z_i), \sum_{i=1}^n I(y_i \neq z_i) \right\}$$

3. Take the matching score of  $x$  to be the maximum of all the matching scores calculated in step 2.

We can interpret this matching score as a potential rate of correct classification. A matching of 1 means that  $x$  can predict  $y$  perfectly, and a rate of 0.5 means that  $x$  is no better than chance.

Figure 2 shows the matching scores of all 256 candidate factors. Most variables are better than chance, and the best have matching scores between 0.7 and 0.8.

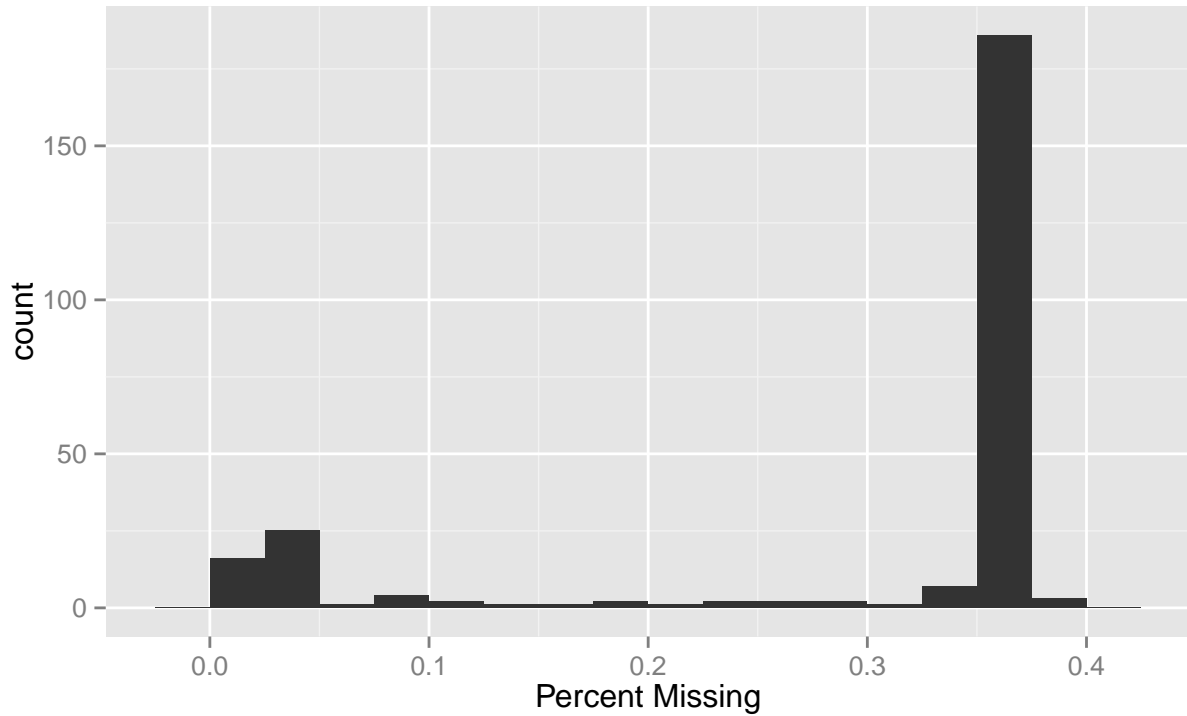


Figure 1: Histogram of the percentage of missing values in each of the available non-null 256 student-specific categorical variables for prediction. Most variables have a large fraction of missing values.

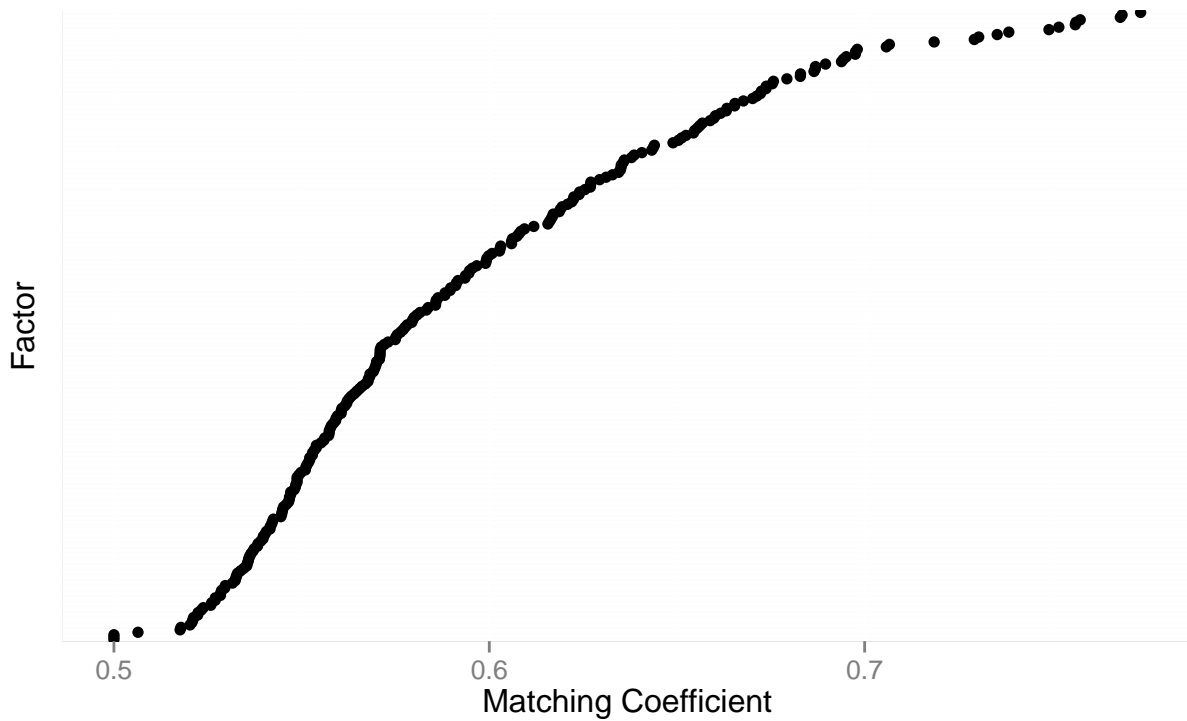


Figure 2: matching scores of all 256 candidate factors. Most variables are better than chance, and the best have matching scores between 0.7 and 0.8.

## Variable selection and preprocessing

Figure 3 shows the matching scores of the top 40 variables. In general, self-efficacy and prior familiarity with the content most closely match outcomes in performance. However, using these variables for prediction would be logically circular and uninformative from a policy standpoint. I remove them from the analysis. Figure 4 shows the variables with which I will build a classifier.

Unfortunately, even with a reduction in the number of variables, there are still a lot of missing values. Figure 5 shows the number of missing values for each student. The figure leads to the following imputation strategy:

1. Remove the students who missed more than 13 questions (only 3.4% of USA students).
2. Put all factors on a numeric scale such that the natural ordering of factor levels is preserved. (All factors are ordinal.) Center and scale the predictor variables, and denote student performance by 1 and -1 for high-performing and low-performing students, respectively.
3. Impute the remaining missing values with nearest neighbor imputation on the 20 predictor variables. I use the `knnImputation` function in the `DMwR` (Torgo 2010) package (setting the number of neighbors to 10).

The resulting dataset looks like:

```
imp = imputedUSA()
dim(imp)
```

```
## [1] 4973    21
```

```
head(imp)
```

```
##      Performance      ST28Q01      ST84Q03      ST43Q06      ST42Q02      ST84Q01
## 1              1 -0.01327612  0.7140237 -1.0157491  0.1777133  0.2515726
## 2             -1 -1.33364707  0.7140237  0.4897006  0.1777133  0.2515726
## 3             -1 -0.01327612  0.7140237  1.0130296  1.2211033  0.2515726
## 4             -1 -0.01327612  0.7140237  0.1344883  0.1777133 -1.0228000
## 5             -1  0.64690935  0.7140237  1.1163704  0.1777133  0.2515726
## 6              1 -0.67346160 -0.9703637 -0.5140267 -0.8656767 -1.0228000
##           HISCED      ST93Q01      ST82Q03      ST94Q06      FISCED      ST42Q01
## 1 -0.783415246 -0.75378499  0.8671840 -0.174306723 -0.3961970  0.6023680
## 2 -0.783415246  0.03564301  0.8671840  0.004407479 -0.8211321 -0.4775471
## 3  0.764457829  0.34645940 -0.1292009 -0.521276932  0.9551568  0.8111194
## 4 -0.783415246  0.19340022  0.8671840 -0.022434066 -0.3961970 -0.4775471
## 5  0.764457829  1.09495985  0.8671840 -0.174306723  0.9551568 -0.4775471
## 6 -0.009478708 -0.26040501 -0.1292009  0.084494718  0.2794799  0.6023680
##           ST88Q01      ST14Q02      ST94Q10      ST91Q06      ST42Q08      ST18Q02
## 1  0.2567285  0.8643092 -0.1989484 -0.08707042  0.0007055227  0.8182283
## 2  1.4239055  0.8643092  0.3201544  1.06411472  0.0007055227  0.8182283
## 3  1.4239055 -1.1567194 -0.1290343  1.06411472  1.1517657877 -1.2218387
## 4  0.2567285  0.8643092 -0.1256085 -0.08707042  0.0007055227  0.8182283
## 5 -0.9104485 -1.1567194 -1.1718213  1.06411472  0.0007055227 -1.2218387
## 6 -0.9104485 -0.1135336  0.4743882 -1.23825556 -1.1503547423  0.4549846
##           ST42Q03      ST79Q03      ST79Q10
## 1  0.2679364  0.7332952 -0.2906867
## 2  0.2679364 -0.2631718  0.7734402
## 3  0.2679364  0.7332952 -0.2906867
```

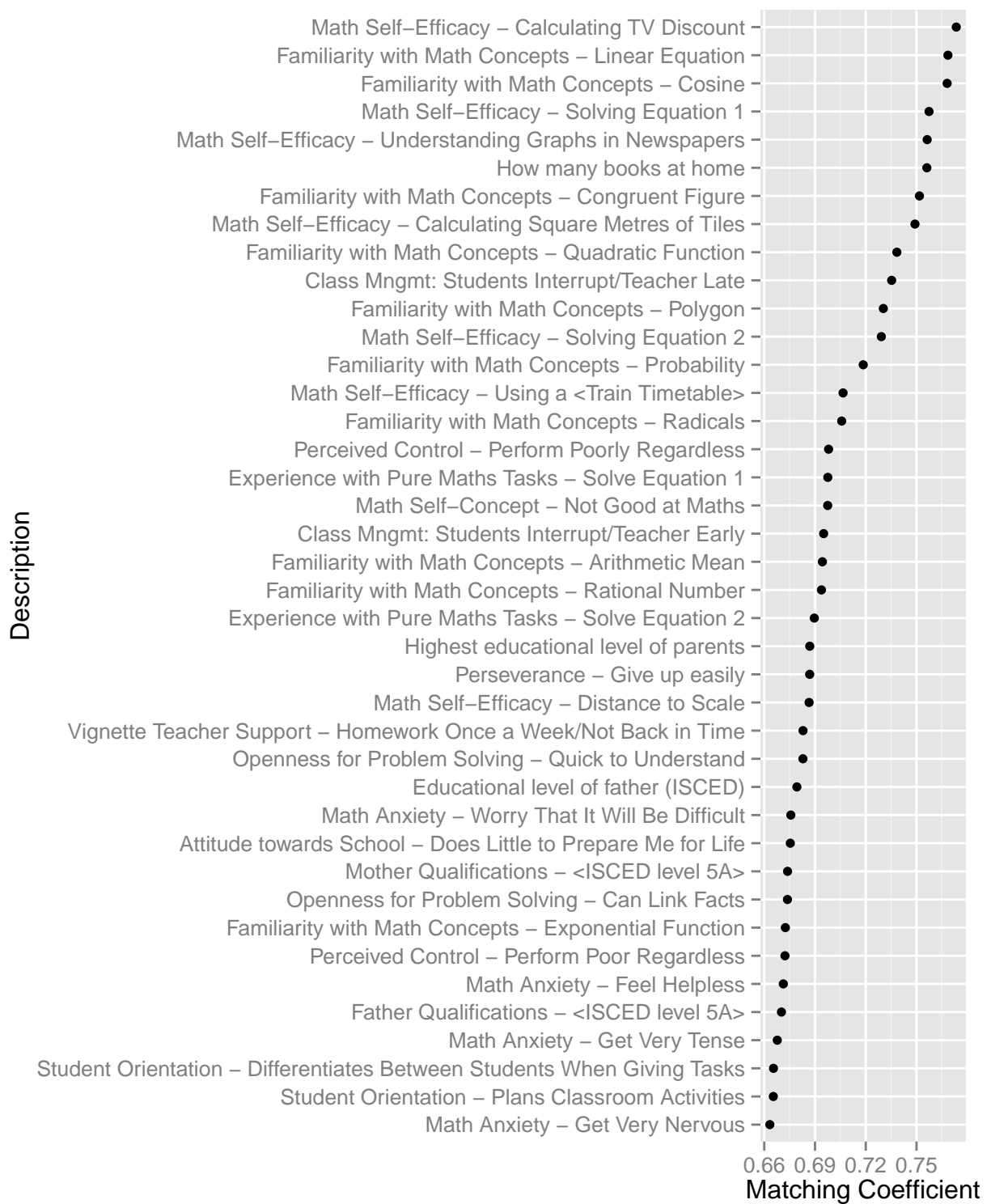


Figure 3:

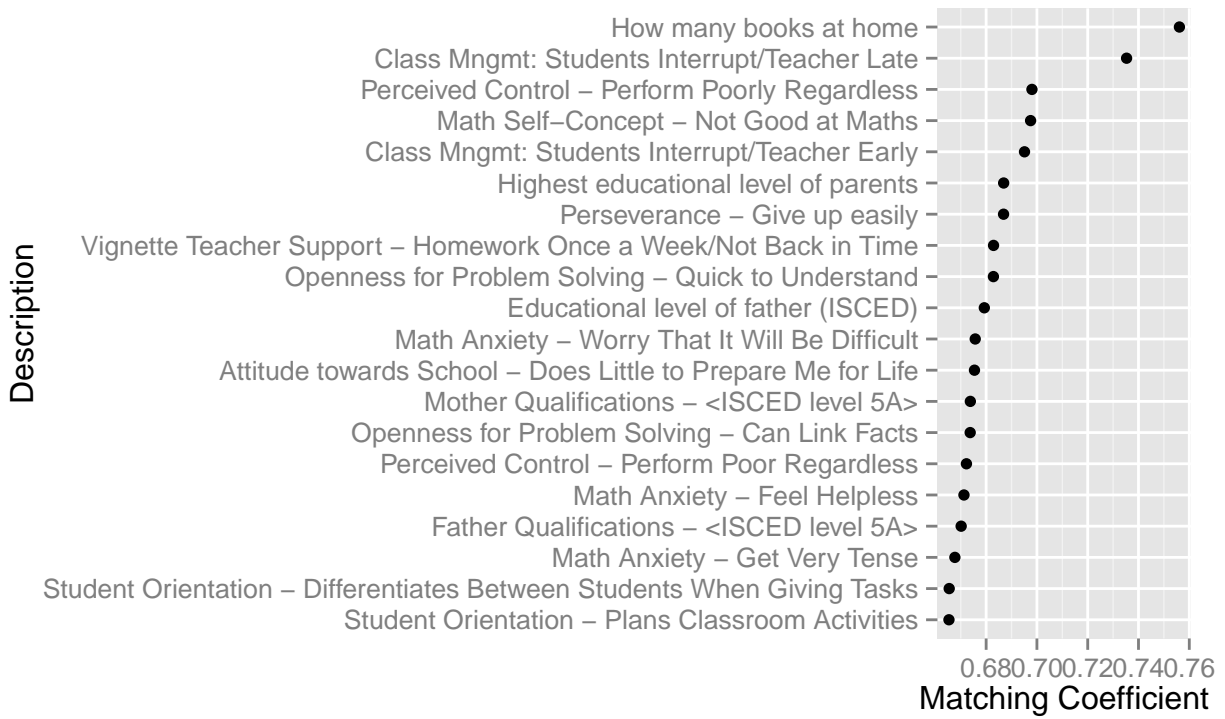


Figure 4:

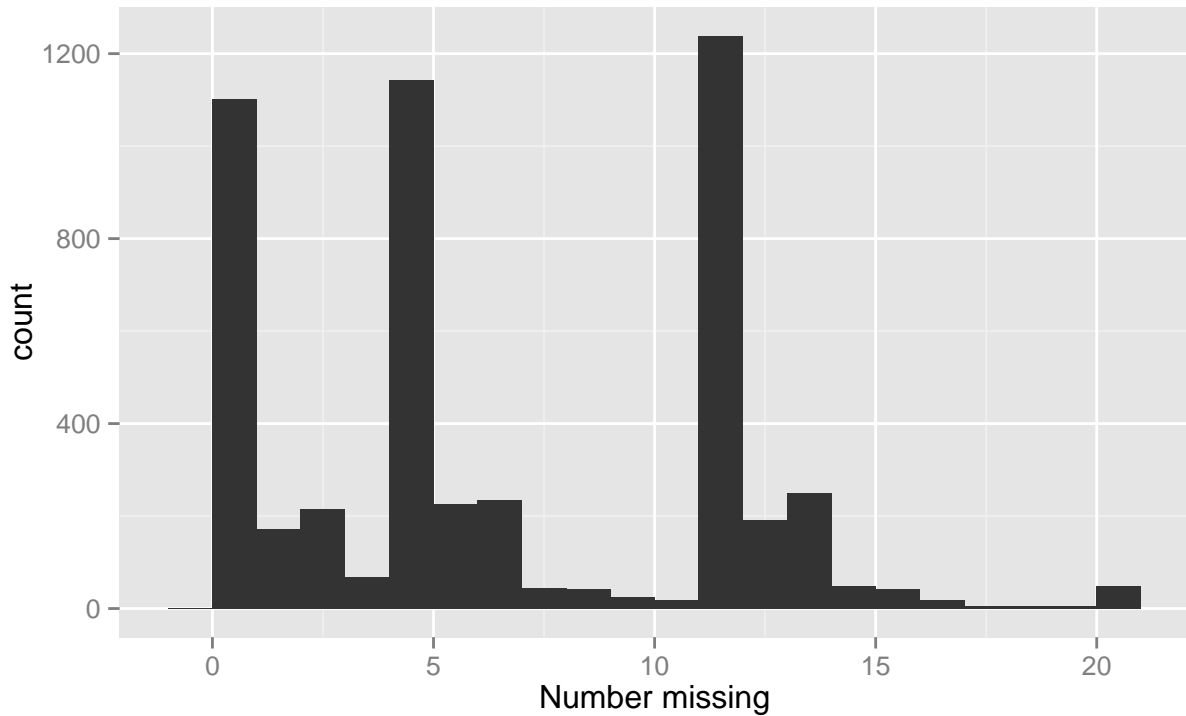


Figure 5: number of missing values for each student. This is after selecting the best 20 variables in the “Variable selection and preprocessing” section.

```
## 4  0.2679364  0.7332952 -0.2906867
## 5 -1.9198197  0.7332952  0.7734402
## 6  0.2679364 -1.2596389 -0.2906867
```

where the 20 predictor variables are defined in the PISA 2012 student dictionary.

## Plan for further work

- Flesh out a more comprehensive coherent story in the exploratory analysis section. I will categorize variables by the major issues to which they pertain and group them by the matching coefficients described previously. This will give me a rough idea of which overall issues (such as parental education, possessions, attitude, etc.) are most important in classifying students.
- Build a classifier on the preprocessed and imputed data using:
  - Logistic regression
  - Neural networks
  - Random forests
  - Nearest neighbors classification
- Run a basic clustering analysis on the 20 predictor variables in the imputed data. I will use kmeans and hierarchical clustering, and I will determine how much information about student performance these techniques recover.

## Acknowledgements

I would like to thank Dr. Cook for steering me in the right direction. The PISA data is messy and cumbersome, and she gave me a much-needed boost.

## References

“Organization for Economic Co-operation and Development.” 2015. <http://www.oecd.org/>.

Torgo, L. 2010. *Data Mining with R, Learning with Case Studies*. Chapman; Hall/CRC. <http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR>.