Jeremiah Herr, Prime Number Analysis 2, DATA 445

Abstract

The goal for this project was to try and create a model that can accurately classify which numbers in a given set are prime and not prime based on a given set of parameters (as opposed to determining primality based on a test). As one might expect, none of the tested models were able to consistently identify true primes.

Introduction

The data used for this project were all the integers 1 through 10,000. Using the Java programs I wrote for the last project (with some modifications), I generated these integers (along with some parameters of interest) and stored them in NumberSample.txt. Then I performed statistical analysis on the data using R (recorded in Nums.R) and stored the important outputs in NumAnalysis.txt.

The data was organized (and labeled in R) as follows: the first column ("nums") is the set of integers 1 through 10,000. The second column ("nth") gives the numbers of primes less than or equal to each number. The third column ("order") gives the order of magnitude of each number (in base 10). The fourth column ("prop") gives the proportion of numbers that are prime at each order of magnitude (these first four parameters are the exact same ones from the previous project and were included for convenience). The fifth column ("gap") gives the difference between each number and the largest prime less than itself (1 and 2 are the only exceptions; 1 was given a gap of -1 and 2 a gap of 0, as there are no positive primes less than either of these two numbers). The sixth column ("mod6") gives the value of each number modulo 6 (since every prime greater than 3 is of the form $6k + 1$ or $6k + 5$). The seventh column ("mod2") gives the value of each number modulo 2 (to determine whether each number is even or odd). The eighth column ("prime") tells whether each number is prime, where "N" means a number is not prime and "Y" means a number is prime.

Methods

Six models were used in this project, namely logistic regression, LDA, QDA, KNN, ridge regression, and the Lasso—in order to practice using each one and to compare their results and determine which (if any of them) could be used classify primes. The only concern in this project was the predictive accuracy of the models, hence their interpretability wasn't considered and no subset selection methods were used. Since the number of parameters was very small relative to the sample size, dimension reduction methods were not used either.

For each model, k-Fold Cross-Validation was used to resample the data and estimate test errors; since LOOCV was computationally expensive for such a large data set (albeit still feasible), and k-Fold CV was easier to manually implement than the Bootstrap (only logistic regression had either of these two methods already implemented in the packages I was using). In each case a value of $k = 10$ was chosen (except for the QDA model, where every $k < 23$ generated an error in R). While applying resampling methods to the data in this case was unnecessary (since it is extremely easy to generate another large sample of integers), they were still used for practice and to demonstrate that I understand how and why to use them.

Each model generated a pair outputs: a table showing the number of accurately and inaccurately classified integers, and a test error estimate (given as the mean of the proportion of misclassified integers). Each table gives the number of true negatives (row N, column N), false positives (row N, column Y), false negatives (row Y, column N), and true positives (row Y, column Y).

Results

The logistic model (Output 1, NumAnalysis.txt) accurately classified 8,776 non-prime integers and 1 true prime and misclassified 869 primes and 354 non-primes. With an estimated test error of 0.1223, we can see that a majority of the integers were accurately classified, however only one of those was prime. Most of the errors were primes being classified as non-prime, though there were still many non-primes being classified as prime. So, the logistic model could classify most non-primes correctly, but struggled to classify true primes. The performance of the logistic model is not surprising since the prime distribution is certainly not logistic and the primes are distributed non-linearly (given the logistic model is comparable to linear models).

The LDA model (Output 2) had results fairly similar to that of the logistic model (the test errors were just about the same), except virtually all the errors were from the model classifying almost every prime as non-prime. Only 5 integers were classified as prime—two of which actually were prime. So, the LDA model tries to classify almost every integer as non-prime and does not appear to be any more or less accurate than the logistical model. The performance of the LDA model is also not surprising here since the primes are neither normally nor linearly distributed.

The QDA model (Output 3) had a very different output. It accurately classified 5,046 non-primes and misclassified the remaining 3,768. So, the QDA model classified the non-primes far less accurately than the other two models (though it still classifies most of them correctly). Interestingly, however, the model classified almost every prime correctly (with the exception of two primes). With an estimated test error of 0.3768, we see that this model is far less accurate overall than the previous two models, but it manages to obtain a far larger proportion of true positives than the others. While the QDA was able to achieve a higher number of true-positives, it came at the cost of a much higher amount of false-positives. I was very surprised by the QDA model classifying so many primes correctly and by the relatively high test error, as I had expected its performance to be more similar to that of the previous two models while being perhaps slightly more accurate. While the assumption of the data following a normal distribution clearly does not hold, the quadratic assumption may be more accurate in some sense than a more linear one.

The KNN model (Outputs 4-7) gave results almost identical to that of LDA. Nearly all of the non-primes were classified correctly, while just about every prime was misclassified. Different values for k did not improve the accuracy of the model in any meaningful way. As k increases, KNN will classify every single integer as non-prime. At first, I found this behavior surprising as I expected the KNN model, due to being more flexible and non-linear, to perform better. After thinking through the nature of the model and of the prime distribution, however, I realized that this was a predictable outcome because the model groups data points according to their proximity to each other (it even says so in its name!), and the primes tend to be spread apart from each other and are instead surrounded by many non-primes. Hence, as more and more neighbors are taken into account, primes are more likely to get grouped with non-primes.

Finally, the ridge regression and Lasso models (Outputs 8 and 9 respectively) were precisely identical in their outputs and both made predictions roughly equivalent to those made by the LDA and KNN models. Given that ridge regression and the Lasso extend the more linear models, I expected them to outperform the logistical and LDA models at least somewhat. However, that does not appear to be the case.

Conclusions

Overall, none of the tested models could reliably classify numbers as being prime or not prime. Each model would greatly underestimate or overestimate the number of primes. This is not surprising as the distribution of primes appears to be very random and there are few (if any) reliable methods for finding primes other than testing various numbers to verify that they do not divide a candidate number.