Robert Stephenson

Dr. Hieu Pham

IS 471-01

17 March 2023

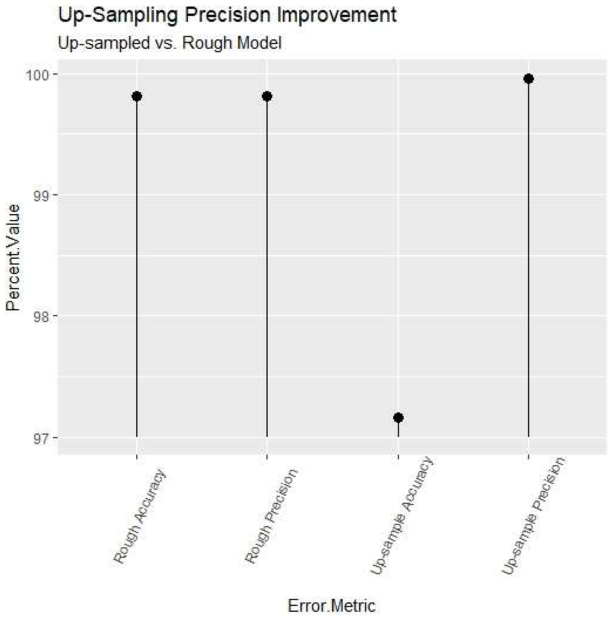
Midterm Write-up

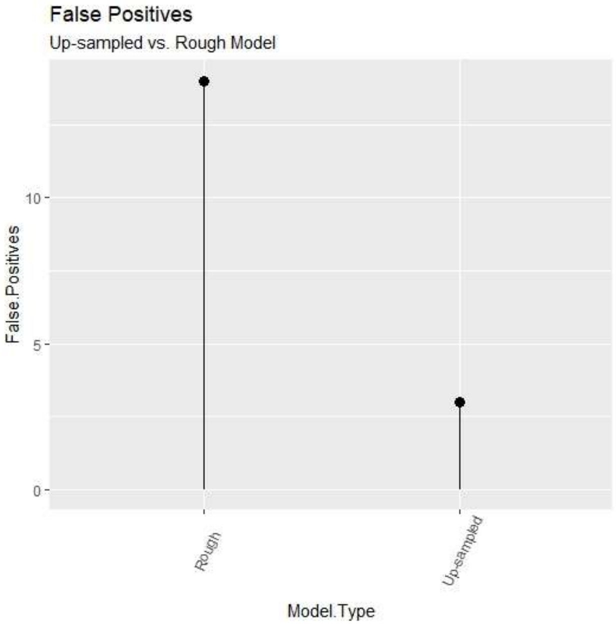
The problem to be solved in this instance was trying to predict which websites are safe and which are unsafe by either labelling them “good” or “bad” so you can increase the security of your computer networks by blocking “bad” websites. For this problem I decided to start developing a logistic regression algorithm. To measure my algorithms effectiveness I chose to measure both accuracy and precision. The reason for using precision as an error metric is because for this problem you want as few false positives (labelling a bad website as good) as possible.

I started by importing the labelled website data and first I checked how many missing values there were and I found out there were seven missing values. Because the amount of missing values was low (7 out of 36622) I just decided to remove them rather than worrying about putting extra time and effort into imputing these missing values. I then converted the “good” and “bad” in the label column to 0 and 1 respectively. I did this because for whatever reason the logistic regression algorithm didn’t like trying to predict on “good” and “bad” and much preferred 0 and 1. I then removed the IP address and unique ID columns as they seemed to be completely unique identifiers and I felt they wouldn’t add any value for our predictions. I didn’t originally plan on removing the website domain column however, it was causing memory issues and made it impossible to try and read the model summary as it presented every possible value the website domain column as a coefficient. I decided to remove the column to be able to make meaningful observations on what coefficients were important, reduce computational stress, and to make the algorithm much more pleasant to deal with it. Since there was also so many unique values of the website domain column, I didn't feel like it would compromise the accuracy of the predictions to treat is as sort of a unique identifier column and therefore remove it.

It seems that the removing of these columns didn't affect the model a great deal because running a logistic regression on this data with an 80/20 split on seed 2112 without any tweaking of any kind resulted in an accuracy and precision of 99.81%. The first or "rough" model however had 14 false positives (with good/0 being a "positive " and bad/1 being a "negative") meaning that 14 actually bad websites would've been labelled as good by the algorithm. While this is still proportionally a really low false positive rate we want it to be as low as possible because when it comes to predicting unsafe websites we would rather be safe than sorry.

I next tried spreading the server location column to create a wider data frame however upon running it through the algorithm I had the exact same results as the first or "rough" algorithm, so it didn't seem to do much of anything at all. I then decided to try a different method and I settled on up-sampling to try and help with class imbalance and possibly reduce the false positive rate a.k.a. increase precision. I was very happy to see that while up-sampling slightly decreased overall accuracy, down to 97.16%, it significantly lowered the amount of false positives to just 3. This increased the precision of the model to 99.96% which is excellent for this problem.





I was highly satisfied with this model as even though it took a hit to accuracy there was significantly less false positives. I found there was no parameter tuning to worry about as I searched and found there were no tunable parameters for a glm() algorithm, so I settled with this algorithm for my best model. I knew however that this couldn’t be my final model I recommend to a firm because it has only been tested once on an independent test set and that would be irresponsible. To fix this problem I ran the up-sampled algorithm 250 times on random independent test sets and had the same results as the first time I tested the model with an accuracy of 97.16% and a precision of 99.96% with only 3 false positives. Knowing that I had a well-tested final model I made predictions on the unlabeled website data and added the predictions as a new column to the data and saved it as “websites_new_append.csv” and then saved “finalGlmStephenson.rda” as my final model. I then compared what the rate of websites predicted as bad was for the new predictions of the unlabeled data. I found that my model labelled 5.65% of the unlabeled websites as bad. I compared this to the labeled websites and found that 2.11% of the labeled websites were bad. This was reassuring because this told me it would be less likely for a bad website to slip through the model’s detection. This will be a highly valuable model for any firm concerned about the safety of their computer networks with its high accuracy and high precision, which in turn means very few unsafe websites (2.07%) avoid detection.