**DATA622: Homework2: Essay**

by Glen Davis

For Homework 2, I chose [a dataset of Web sites](https://www.kaggle.com/datasets/danielfernandon/web-page-phishing-dataset) labeled either Phishing or Legitimate. In addition to this binary response variable, there were 19 integer predictor variables. Most of them represented counts of specific punctuation characters within the Web sites’ urls, like slashes or dots, and there was also a count of the total number of characters within the url, as well as a count of redirects within the url.

Many of these predictors demonstrated near-zero variance and would only have served as noise in any models I built, so I removed them from consideration. The five predictors I was left with were the counts for: dots, hyphens, slashes, redirects, and total characters. Of the three models I built, none of them ended up using the count of redirects.

The two predictors that were most correlated with the response variable were the count of slashes and the count of total characters, which were also pretty correlated with each other. When building the decision tree models, I excluded the count of total characters from the first model, and I excluded the count of slashes from the second model since a single decision tree trained on both features might not be able to use the information from both of them anyway. With two correlated predictors, whatever feature is not used for an early split isn’t typically going to be very useful in creating a later split because it just reinforces what we already know.

Indeed, the accuracy of both decision tree models was similar despite the fact that each of them was missing one of the strongest predictors of Phishing Web sites. However, each decision tree model suffered more from one kind of classification error than the other. Decision Tree Model 1 had higher precision, so it was better at limiting the number of irrelevant Phishing alerts. Relatively few of the positive alerts it generated were false positives. On the other hand, Decision Tree Model 2 had higher recall, so it captured most of the Web sites that were actually Phishing Web sites. It classified relatively few Phishing Web sites as Legitimate Web sites.

If my business had to choose between the decision tree models only, there is a clear trade-off in how the models perform, and I think my preference would probably be for Decision Tree Model 2. It is the safer of the two, in that deploying it would prevent employees from being exposed to more Phishing Web sites. Although it might annoy employees because a lot of Legitimate Web sites would be blocked as well, it’s easier to over-censor at first and slowly add Web sites to a “cleared” list than it is to under-censor at first and incur more harm.

One of the decision tree concerns listed in the blog is logic repetition. While my decision trees wouldn't need to be manually maintained, and my concerns are therefore different from the blog author's, features were in fact used multiple times in my decision tree models to make different splits. That can generally be confusing to users, and I find myself reading a decision tree much more slowly when the same feature is used again and again throughout it. Another concern listed in the blog is complexity, which my decision tree models didn’t really suffer from because they weren’t very deep. If they had been deeper, the logic repetition concern could definitely have been exacerbated, and I might have pruned the decision trees further than recommended for the sake of readability. The last concern the blog mentions that applies to my trees is familiarity. Since every predictor variable was a count, my features weren’t at all obscure, so my decision trees are thankfully quite approachable. The less intuitive a predictor is though, the more trouble a user is going to have interpreting the decision tree, so care has to be taken there.

In reality, my business would rarely be forced to choose between my decision tree models though. By aggregating a variety of decision trees using any two of the predictors, the Random Forest Model overcame some of their limitations. The increased accuracy and balance between precision and recall I saw made it an excellent Phishing Web site classifier. What’s even more impressive is that the random forest model could probably have identified the important predictors and generated pretty good predictions even if I hadn’t trimmed the feature space myself during exploratory data analysis. While we should always do the work of getting rid of any noise we can identify, random forest models aren’t as sensitive to it as some other models. I really appreciate having the ability to extract relative feature importance estimates from these models. Even though you can’t visualize them like decision tree models, you can still see what irrelevant features they weeded out on their own and what features they considered most important.