Phishing Classification

Evan Reed | ereed30@gatech.edu | GT ID: 480

17 April 2022

ISYE 7406 Data Mining & Statistical Learning: Course Project Report

◊ ABSTRACT

Phishing is a common place cyberattack used by cyber criminals to trick targets into providing access to their computer systems and data. The exact specifics of how these attacks are conducted vary in complexity, but the general goal always remains the same: trick a target into willingly clicking on a link they think is for something else. When this goal is accomplished, the victim will have just unknowingly provided the attacker access to their system and data. So how do we protect ourselves when criminals are constantly devising more believable and elaborate phishing schemes?

◊ INTRODUCTION

Effectively identifying malicious webpages before potential victims have time to unknowingly click into them is a great first line of defense against phishing attacks. This report explores the possibility of identifying malicious webpages using general technical characteristics that comprise every webpage in existence. Using these metadata characteristics, data mining and statistical learning techniques are explored to determine if a webpage can be classified as either malicious or legitimate.

◊ DATA SOURCE

The data source for this analysis contains details of 10,000 webpages, which were identified as either malicious or legitimate with an even split of 5,000 for each category. In addition to the classifier variable (malicious vs legitimate), each webpage has 48 features that were captured through a browser automation framework used to parse and collect this information.

Each of the 48 features could be broken down into 4 data type groups:

Data Type	Number of Features
Binary, (0,1)	23
Integers	16
(-1, 0, 1)	6
Percentages	3

Here is the full list of features and their data types:

Feature Name	Data Type	Feature Name	Data Type	Feature Name	Data Type
NumDots	Integer	IpAddress	Binary	AbnormalFormAction	Binary
SubdomainLevel	Integer	DomainInSubdomains	Binary	PctNullSelfRedirectHyperlinks	Percentage
PathLevel	Integer	DomainInPaths	Binary	FrequentDomainNameMismatch	Binary
UrlLength	Integer	HttpsInHostname	Binary	FakeLinkInStatusBar	Binary
NumDash	Integer	HostnameLength	Integer	RightClickDisabled	Binary
NumDashInHostname	Integer	PathLength	Integer	PopUpWindow	Binary
AtSymbol	Binary	QueryLength	Integer	SubmitInfoToEmail	Binary
TildeSymbol	Binary	DoubleSlashInPath	Binary	IframeOrFrame	Binary
NumUnderscore	Integer	NumSensitiveWords	Integer	MissingTitle	Binary
NumPercent	Integer	EmbeddedBrandName	Binary	ImagesOnlyInForm	Binary
NumQueryComponents	Integer	PctExtHyperlinks	Percentage	SubdomainLevelRT	(-1, 0 1)
NumAmpersand	Integer	PctExtResourceUrls	Percentage	UrlLengthRT	(-1, 0, 1)
NumHash	Integer	ExtFavicon	Binary	PctExtResourceUrlsRT	(-1, 0, 1)
NumNumericChars	Integer	InsecureForms	Binary	AbnormalExtFormActionR	(-1, 0, 1)
NoHttps	Binary	RelativeFormAction	Binary	ExtMetaScriptLinkRT	(-1, 0, 1)
RandomString	Binary	ExtFormAction	Binary	PctExtNullSelfRedirectHyperlinksRT	(-1, 0, 1)

As seen by the list above none of the variables are related to the content or sentiment of the webpage, but rather the general technical characteristics.

◊ PROPOSED METHODOLOGY

The methodology needed for creating an effective classification model can be broken into the following steps.

1. Exploratory Data Analysis – Understand the dataset
2. Identify Possible Models – Decide appropriate models for the problem
3. Model Exploration and Selection – Compare the models to find the best fit
4. Model Tuning – Further improve the model selected if possible

Firstly, exploratory data analysis is needed to gain an understanding of the underlying data that would be crucial for producing the final model. In this step the features are analyzed at an individual level to determine if each variable alone is fit to be included in the model. Features with minimal variability (i.e., most values are the same across all webpages) would be removed from the dataset as they have no relation to the classification. Next the features need to be analyzed to identify high levels of correlation between the features. Using this information, redundant features can be removed to simplify the model and reduce the amount of multicollinearity.

Once an understanding of the dataset is established, the next step is selecting the appropriate model to classify the webpages. At a very high level, the problem can be described as needing a supervised classification model. Classification as we are trying to group the data into a specific category, specifically classifying malicious vs legitimate webpages, and supervised as the true classification of each webpage it is known when training the model. This enables a direct way to calculate to accuracy of the output of the model.

With this understanding, the following models are appropriate for this problem:

- Random Forest
- Boosting
- Linear Discriminant Analysis (LDA)

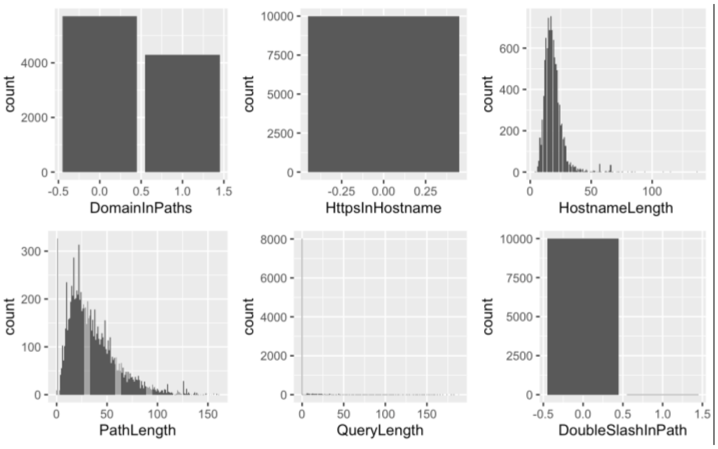
- Naïve Bayes
- Logistic Regression
- K Nearest Neighbors (with K = 3, 5, 7, 9, 11)

Once appropriate models have been identified, the next step is training the models to determine which one is the best fit. This is done through the process of cross-validation, which reduces the effect of randomness in training a model to avoid an inferior model being selected as the best one. Cross validation trains multiple versions of the same models only each time using a different subsection of the dataset to train and test the model.

After deciding the best fitting model from the cross-validation results by calculating the highest accuracy of correctly classifying a webpage as malicious or legitimate, the final step is tuning the input parameters of the model to further improve its accuracy.

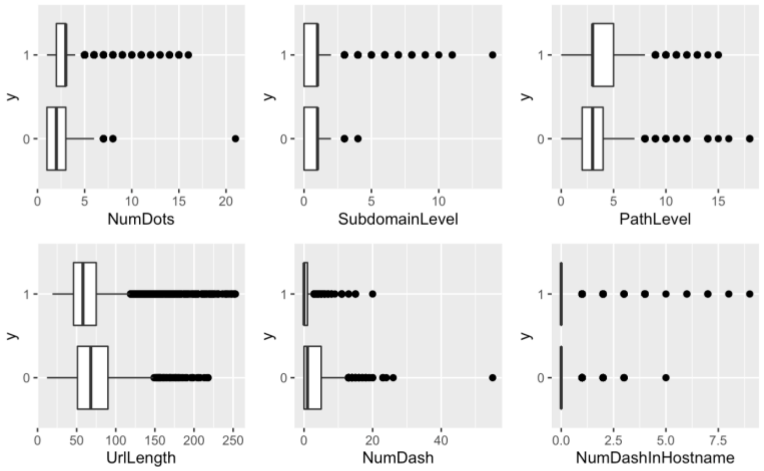
◊ ANALYSIS / RESULTS

Using the methodology described above, the analysis began with exploratory data analysis of the dataset. First a histogram of each variable was constructed.



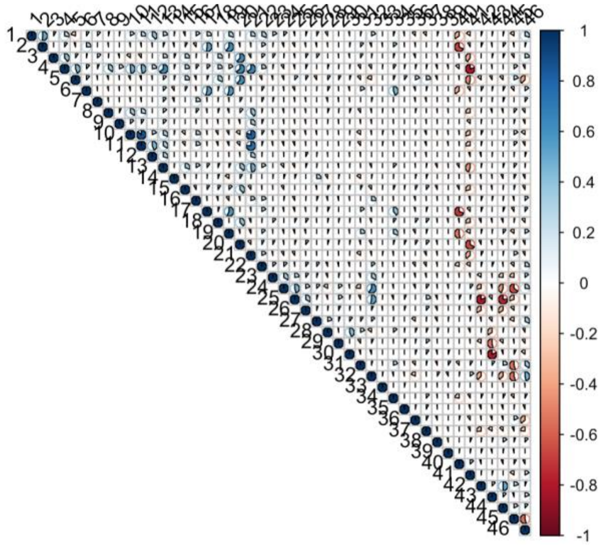
Analyzing the histograms of each of the 48 features identified 3 features that could be removed from the set of predictors due to their lack of variability within both classifications: AtSymbol, HttpsInHostname, and DoubleSlashInPath.

Next, the exploratory data analysis continued by plotting boxplots of each of the remaining 45 features split by classification.



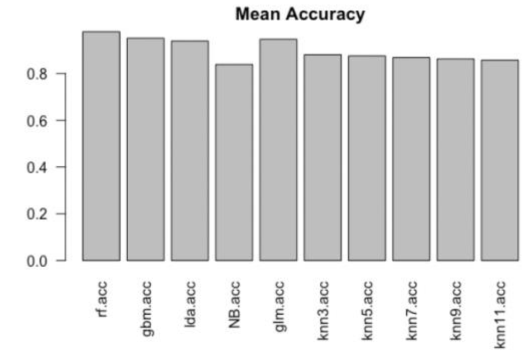
The intent of these plots was to identify any features that exhibited the same distributions between the classification groups, which would indicate the features inability to distinguish between the two groups. However, analyzing the results of each was inconclusive and all 45 features remained in the dataset.

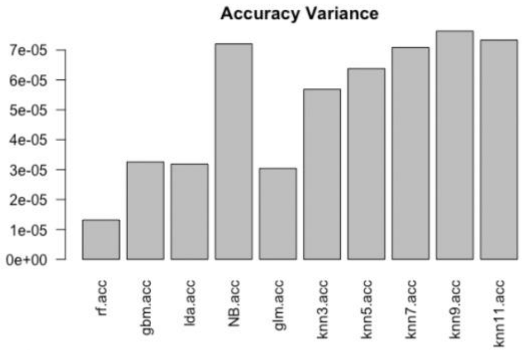
The final step in the exploratory data analysis was examining the correlation between the features.



The correlation matrix above highlighted several variables that were both strongly and negatively correlated; however, none had an absolute value greater than 0.9, so it was decided to keep all 45 remaining features in the dataset.

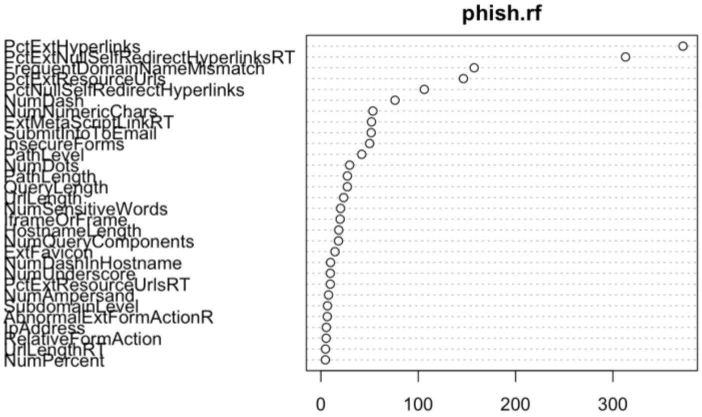
With the data features decided, the next step was the model creation and selection process. First, the data was split into training and testing subsections of the entire dataset using a 70/30 split. Next, the 11 models defined in the proposed methodology section were trained within a cross-validation exercise. Using the 7,000 webpages from the training data 100 iterations were conducted, each iteration using randomly selected 5,500 webpages for the training and the remaining 1,500 webpages for testing the accuracy of the model. After cross-validation, the mean accuracy and variances for each model were calculated.





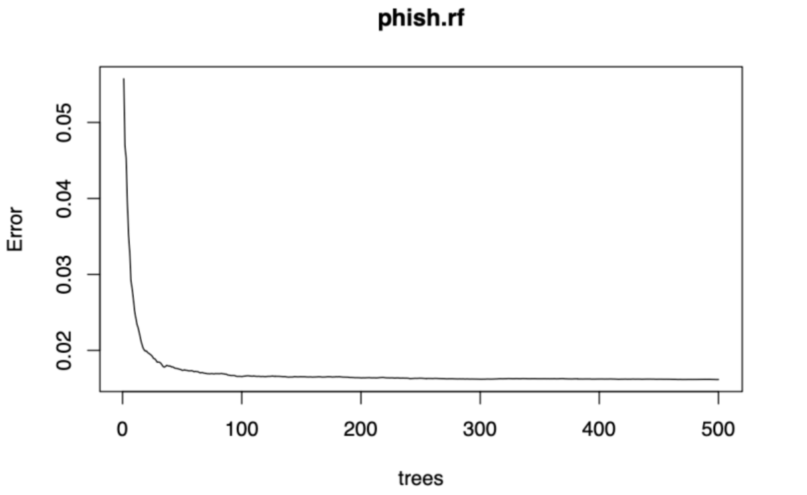
Analyzing the results, the random forest was the best performing classifier. It produced the highest accuracy over the 100 iterations, while simultaneously having the lowest variance.

After selecting the random forest classifier as the best fit for this problem of classifying webpages as malicious or legitimate, further attempts were made to improve the model by tuning the hyper parameters of the model. First the variables used in the model were explored.   Using the rankings from the variable importance plot, additional random forest models were explored.



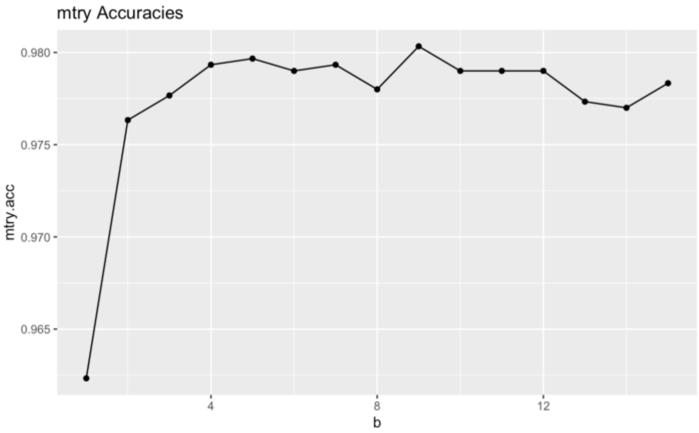
The models using the subsection of the more important variables resulted in a decreased accuracy when the new model was applied to the testing set of 3,000 webpages. It was decided to keep the full feature set that was used in the initial model.

Next, the number of potential trees in the model were compared to the error each variation produced.



This revealed that the error vs the number trees appears to level off around 100, but the lowest error occurs at ntrees of 500. This will be the tuned parameter value used for the random forest model.

Next the mtry parameter was tuned. Different values of the parameter were used to create multiple variations of the model. Then the accuracy for each model was calculated.



The plot above shows that the highest accuracy occurs when mtry is 9. This will be the parameter value for the random forest model.

Finally, the model was tweaked to address the problem at hand. When it comes the classifying phishing webpages it is more important to correctly classify a malicious webpage than it is the mistakenly classify a legitimate webpage. To error on the side of caution, the model was adjusted to classify anything with a value of 0.3 or higher as malicious compared to the previous level of 0.5. This shift in the classifier threshold resulted in a minimal decrease (-.08%) in the accuracy of the model. However, the decrease in accuracy was worth it, as the number of malicious webpages incorrectly identified as legitimate decreased by 56% while only increasing the total number of misclassifications by 24 (89 vs 65) when was applied to the testing set of 3,000 webpages.

◊ CONCLUSIONS

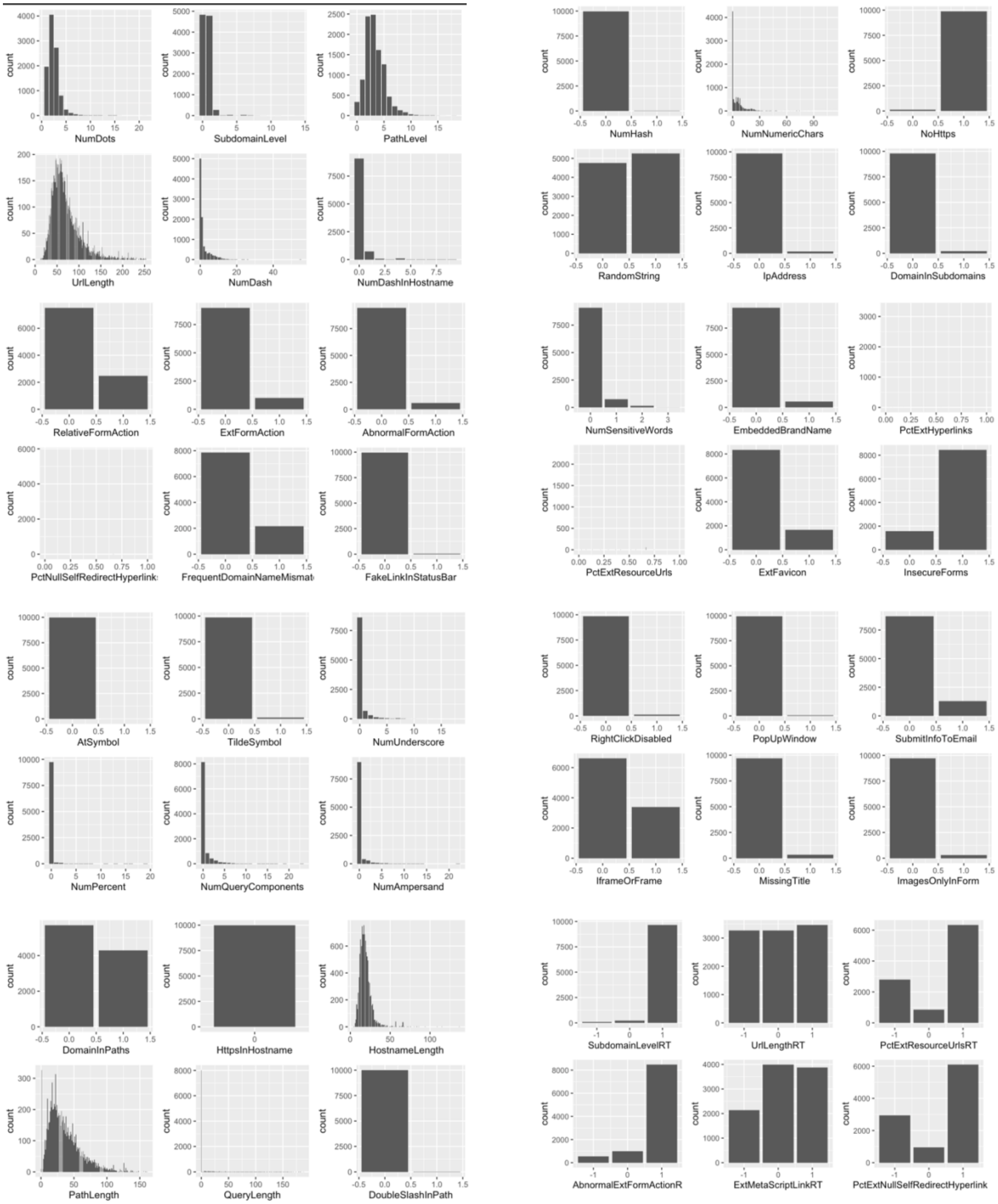
In conclusion, creating a classification model from the features collected using a browser automation framework proved to be highly effective method of identifying malicious webpages. Following the process of exploratory data analysis, model selection, and model tuning, a final model was created the was able to correctly classify 97% of the 3,000 testing dataset webpages. As cyber criminals continue to expand their attacks, there will always be a counter effort to provide protection the population. This project has shown the ability of using current data collection tools and applying data mining and statistical leaning techniques to construct a classification model. The next step and possibly bigger challenge would be constructing a real time implementation of a classifier for users to have on their computers to notify of potential malicious webpages.

◊ LESSONS LEARNED

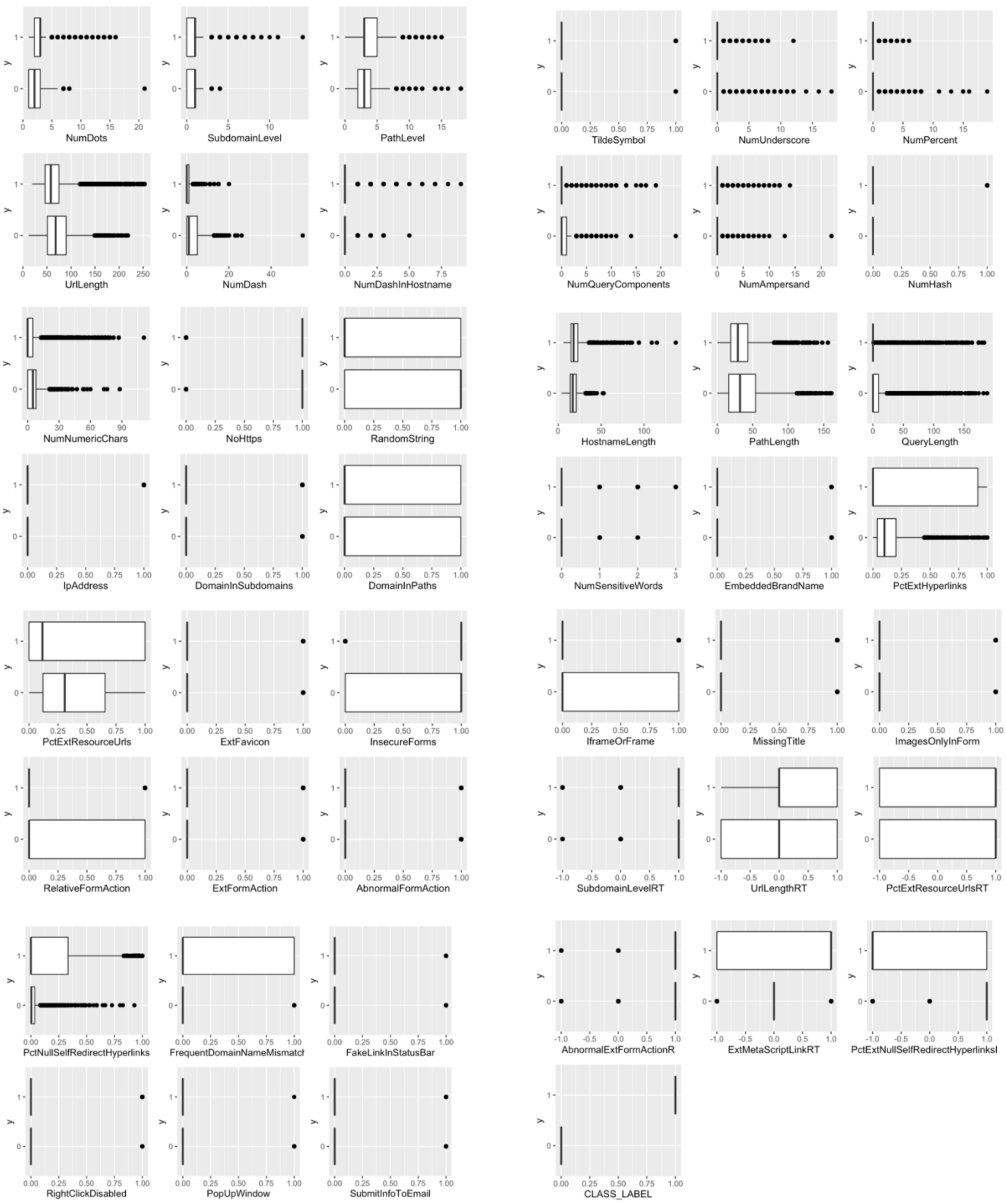
I really enjoyed this class. It provided a free form learning experience, where one is encouraged to explore the material as opposed to being asked to deliver an exact replica of an answer key to get a solid grade. This structure allowed me to think in my own way and practice various types of data mining and statistical learning methods in way that will stick with me more than rote memorization of these methods.

◊ APPENDIX

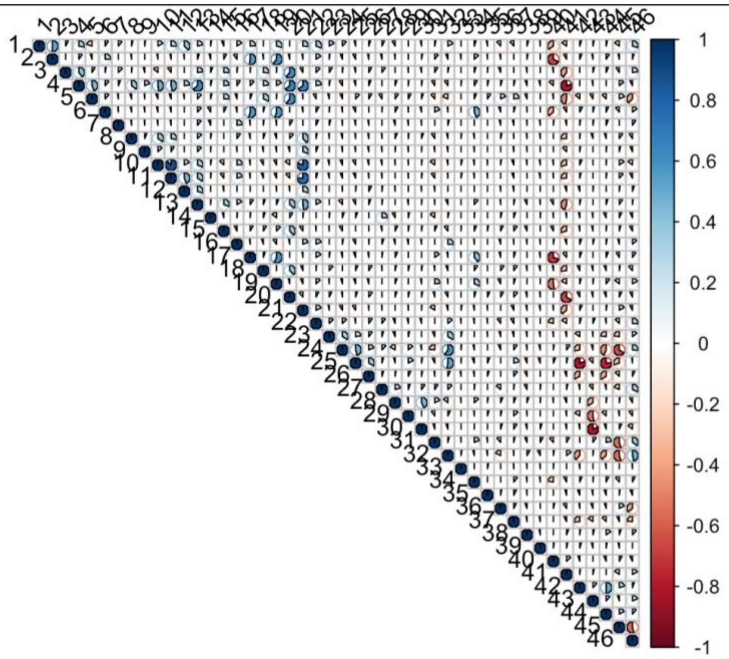
- Histograms of Features



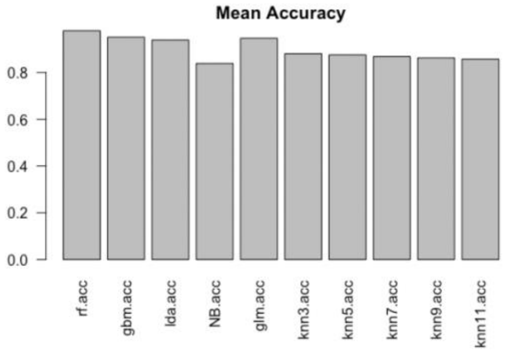
- Boxplots of Features vs Classifier

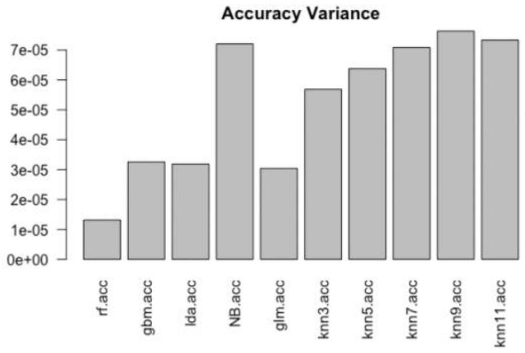


- Correlation Plot



- Cross Validation Plots

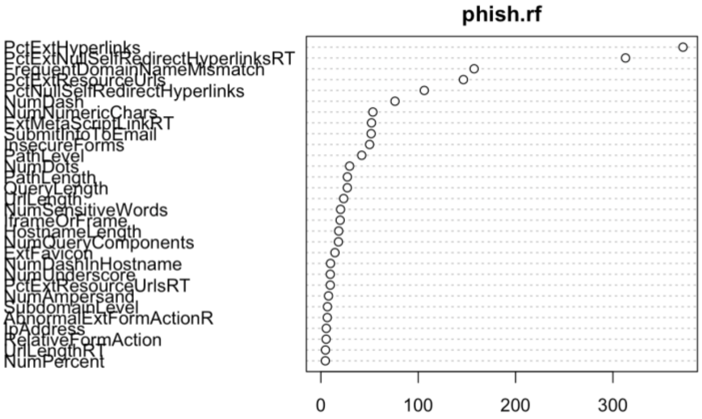


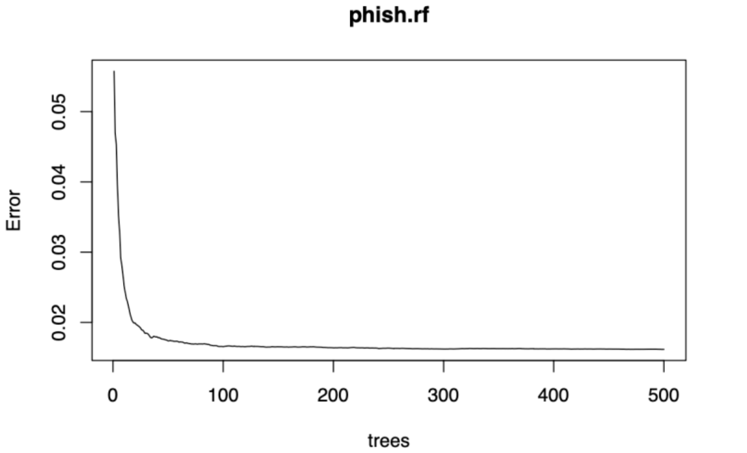


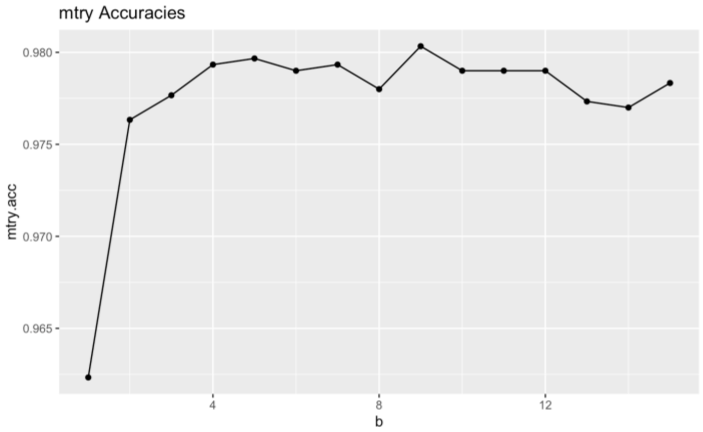
- Table of Cross Validation Results

Model	Mean	Variance
Random Forest	0.9785067	1.31638E-05
Boosting	0.9508533	3.25664E-05
LDA	0.9388333	3.18418E-05
Naïve Bayes	0.8385733	7.2081E-05
Logistic Regression	0.9465267	3.04224E-05
KNN 3	0.8803333	5.68664E-05
KNN 5	0.87536	6.37748E-05
KNN 7	0.8681133	7.08512E-05
KNN 9	0.8626	7.63232E-05
KNN 11	0.85718	7.33186E-05

- Model Tuning Plots







Final Model Outputs

- Before shifting threshold

		Model	
		0	1
Acutal	0	1465	29
	1	36	1470

Accuracy: 0.9783

- After shifting threshold

		Model Output	
		0	1
Actual	0	1421	73
	1	16	1490

Accuracy: 0.9703

- ## Data Source

https://www.kaggle.com/datasets/shashwatwork/phishing-dataset-for-machine-learning