Spam Email Detection

Parker Lutz

University of Central Florida MSDA

Data Mining I Final Project

Executive Summary

In their annual Data Breach Investigations Report, Verizon analyzes the new and evolving patterns in data breaches. The 2016 DBIR outlined a three-pronged attack in which a hacker sends an email containing a malicious link or attachment, the user clicks and downloads malware onto their PC, and the malware proceeds to steal information— like trade secrets or personal passwords—stolen passwords can then be harvested for attacks on third-parties.

In 2015, a cyber-attack against Anthem insurance exposed 78.8 million consumer records after one user opened a phishing email and the hackers gained access to the internal data warehouse.

The 2017 report confirmed that phishing emails are still a threat – they are the most commonly used delivery method for malware. They also found that 1 in 14 phishing emails result in the user clinking the malicious link or attachment. Succumbing to these attacks poses large financial and reputational risk to businesses, giving Spam Email Detection major material importance. It is best to stop these attacks before the end user has an opportunity to fall victim.

In this report, we will use four classification techniques to build a spam email filter. We pay attention to overall accuracy as well as the optimization of false positive and false negative rates, as it is highly undesirable to lose important communications to a spam folder.

Data Analysis

We want to determine if an email is spam. The spam email dataset, available at the UCI Machine Learning Repository, was collected in 1999 and contains 57 features and 1 binary classification. The features assess word frequency, character frequency, and pattern length. For example, the sentence "Click the link below to WIN $1,000 FOR FREE!!!" would be tagged for the word "free" and the sequence of "000" in the monetary value as well as the frequency of exclamation marks would all be accounted for. The run_length variables would also note the sequence of all capital letters.

We can begin the evaluation by looking at the structure of the dataset.

```
> str(data)
'data.frame':   4601 obs. of  58 variables:
 $ word_freq_make            : num  0 0.21 0.06 0 0 0 0 0 0.15 0.06 ...
 $ word_freq_address         : num  0.64 0.28 0 0 0 0 0 0 0 0.12 ...
 $ word_freq_all             : num  0.64 0.5 0.71 0 0 0 0 0 0.46 0.77 ...
 $ word_freq_3d              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ word_freq_our             : num  0.32 0.14 1.23 0.63 0.63 1.85 1.92 1.88 0.61 0.19 ...
 $ word_freq_over            : num  0 0.28 0.19 0 0 0 0 0 0 0.32 ...
 $ word_freq_remove          : num  0 0.21 0.19 0.31 0.31 0 0 0 0.3 0.38 ...
 $ word_freq_internet        : num  0 0.07 0.12 0.63 0.63 1.85 0 1.88 0 0 ...
 $ word_freq_order           : num  0 0 0.64 0.31 0.31 0 0 0 0.92 0.06 ...
 $ word_freq_mail            : num  0 0.94 0.25 0.63 0.63 0 0.64 0 0.76 0 ...
 $ word_freq_receive         : num  0 0.21 0.38 0.31 0.31 0 0.96 0 0.76 0 ...
 $ word_freq_will            : num  0.64 0.79 0.45 0.31 0.31 0 1.28 0 0.92 0.64 ...
 $ word_freq_people          : num  0 0.65 0.12 0.31 0.31 0 0 0 0 0.25 ...
 $ word_freq_report          : num  0 0.21 0 0 0 0 0 0 0 0 ...
 $ word_freq_addresses       : num  0 0.14 1.75 0 0 0 0 0 0 0.12 ...
 $ word_freq_free            : num  0.32 0.14 0.06 0.31 0.31 0 0.96 0 0 0 ...
 $ word_freq_business        : num  0 0.07 0.06 0 0 0 0 0 0 0 ...
 $ word_freq_email           : num  1.29 0.28 1.03 0 0 0 0.32 0 0.15 0.12 ...
 $ word_freq_you             : num  1.93 3.47 1.36 3.18 3.18 0 3.85 0 1.23 1.67 ...
 $ word_freq_credit          : num  0 0 0.32 0 0 0 0 0 3.53 0.06 ...
 $ word_freq_your            : num  0.96 1.59 0.51 0.31 0.31 0 0.64 0 2 0.71 ...
 $ word_freq_font            : num  0 0 0 0 0 0 0 0 0 0 ...
```

Figure 1 snip of data structure output

We can easily see the datatypes of the variables and change the classification label "y" to a factor. Since there are 58 dimensions, it is difficult to visualize the correlations and density values like we could with a lower dimension dataset. We can look at a summary() of the data to quickly assess the mean and distribution of each individual variable.

```
> summary(data)
 word_freq_make    word_freq_address word_freq_all     word_freq_3d       word_freq_our
 Min.   :0.0000    Min.   : 0.000    Min.   :0.0000    Min.   : 0.00000   Min.   : 0.0000
 1st Qu.:0.0000    1st Qu.: 0.000    1st Qu.:0.0000    1st Qu.: 0.00000   1st Qu.: 0.0000
 Median :0.0000    Median : 0.000    Median :0.0000    Median : 0.00000   Median : 0.0000
 Mean   :0.1046    Mean   : 0.213    Mean   :0.2807    Mean   : 0.06542   Mean   : 0.3122
 3rd Qu.:0.0000    3rd Qu.: 0.000    3rd Qu.:0.4200    3rd Qu.: 0.00000   3rd Qu.: 0.3800
 Max.   :4.5400    Max.   :14.280    Max.   :5.1000    Max.   :42.81000   Max.   :10.0000
```

Figure 2 snip of data summary output

It seems that some variables have zero or almost zero variance, which is important to keep in mind throughout the modeling process.

Modeling Approach

K Nearest Neighbors. KNN is a method of classifying test observations by sampling from the k- closest neighboring trained observations in a feature space. We trained a model using 10-fold cross validation and testing for k values from 1 to 40. The model attains highest accuracy at k=2.

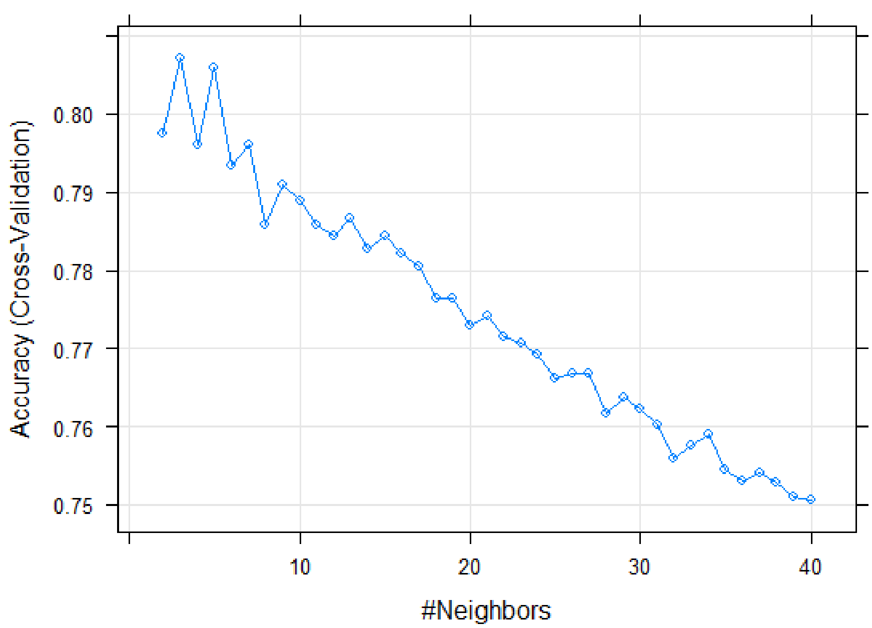

Figure 3 tuning k for knn

We made a prediction on the test set of 601 observations using the trained model and achieved 88.5% accuracy.

	Not Spam	Spam
Pred: Not Spam	344	39
Pred: Spam	30	188

Table 1 knn confusion matrix

	Performance
False Positive	13.8%
False Negative	10.2%
Accuracy	88.5%

Table 2 knn accuracy

Linear Discriminant Analysis. LDA is a method of classification which reduces the dimensionality of the dataset into a linear combination of the features which explain the differences in the predicted classes. It then computes a linear boundary between classes.

The prediction on test data for LDA yields 87.2% accuracy.

	Not Spam	Spam
Pred: Not Spam	349	52
Pred: Spam	25	175

Table 3 lda confusion matrix

	Performance
False Positive	12.5%
False Negative	13.0%
Accuracy	87.2%

Table 4 lda accuracy

Quadratic Discriminant Analysis. QDA is a model that computes a quadratic boundary between classes. Since QDA works by computing a separate covariance matrix for each class, we narrowed the training set down to features that do not have zero or near zero variance before training. The prediction on the test set has 72.5 % accuracy. While QDA is more flexible than LDA, it may have low accuracy because of high model bias if the boundary is truly linear.

	Not Spam	*Spam*
Pred: Not Spam	357	148
Pred: Spam	17	79

Table 5 qda confusion matrix

	Performance
False Positive	17.7%
False Negative	29.3%
Accuracy	72.5%

Table 6 qda accuracy

Logistic Regression. Logistic regression computes the log-shaped line of best fit for the data by assigning coefficients to the individual feature inputs in the model.

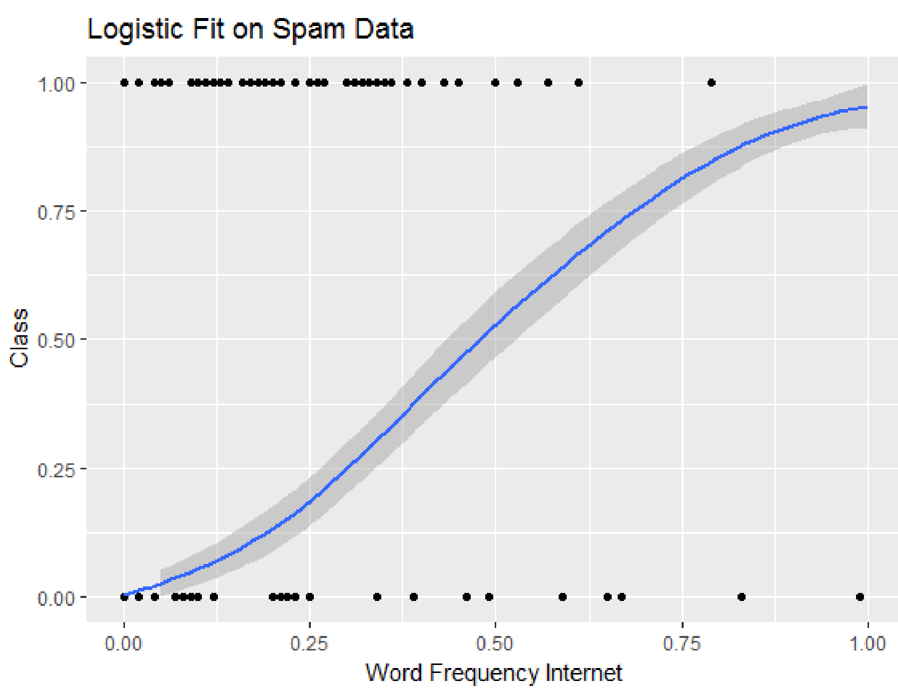

Figure 4 logistic regression line

Our prediction on a test set yields 91.7% accuracy, which makes logistic regression the “winning” model approach.

	Not Spam	*Spam*
Pred: Not Spam	345	32
Pred: Spam	18	206

Table 7 logistic regression confusion matrix, threshold = 0.50

	Performance
False Positive	8.0%
False Negative	8.5%
Accuracy	91.7%

Table 8 logistic regression accuracy, threshold = 0.50

Logistic Regression has the highest accuracy of the four modeling approaches, so it is the model we chose to make a deeper dive on. One of the key goals of this task is not only to achieve high accuracy and limit the number of spam emails that pass through the filter, but also NOT to classify normal emails as spam—so that potentially important communications are caught in a junk folder. The generalized linear model function in R, using the binomial family, predicts a class by predicting the probability of belonging in class 1 and using a threshold of 0.50 so that every observation with probability greater than 0.5 is classified as class 1. It is possible for us to update the threshold to achieve the desired result of zero or near-zero real emails being misclassified as spam.

	Not Spam	*Spam*
Pred: Not Spam	349	37
Pred: Spam	14	201

Table 9 logistic regression confusion matrix, threshold = 0.60

	Performance
False Positive	6.5%
False Negative	9.6%
Accuracy	91.5%

Table 10 logistic regression accuracy, threshold = 0.60

	Not Spam	*Spam*
Pred: Not Spam	353	51
Pred: Spam	10	187

Table 11 logistic regression confusion matrix, threshold = 0.70

	Performance
False Positive	5.1%
False Negative	12.6%
Accuracy	89.9%

Table 12 logistic regression accuracy, threshold = 0.70

	Not Spam	*Spam*
Pred: Not Spam	363	124
Pred: Spam	0	114

Table 13 logistic regression confusion matrix, threshold = 0.97

	Performance
False Positive	0.0%
False Negative	25.5%
Accuracy	79.4%

Table 14 logistic regression accuracy, threshold = 0.97

We can increase the probability threshold the 0.97 and achieve a 0% false positive rate—but 25.5% of spam emails will slip through the filter. It is a business decision which of the false positive rates is acceptable, but setting the threshold at 0.60 which achieves a 91.5% accuracy with only a 6.5% rate of misclassifying normal emails seems like an optimal choice.

Results and Conclusions

Logistic regression is the best choice for this binary classification task. As explored in the modeling stage, we can increase the probability threshold the 0.97 and achieve a 0% false positive rate—but 25.5% of spam emails will slip through the filter. It is a business decision which of the false positive rates is acceptable, but setting the threshold at 0.60 which achieves a 91.5% accuracy with only a 6.5% rate of misclassifying normal emails seems like an optimal choice.

It is also important to note that we achieve 39.6% accuracy by simply classifying every email as spam, and 60.4% accuracy by classifying every email as normal. The optimization of false positive and false negative rates is a key component of this task.

For further analysis, it would be wise to try Principle Component Analysis or some other effort to reduce the number of features being input into each model.

It is also pertinent to update the dataset so that the features correlate to new trends in phishing attempts.

Bibliography

Lichman, M. (2013). UCWEMachine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Mukherjee, S. (2017, January 09). Foreign Government Likely Behind Huge 2015 Anthem Data Breach: Report. Retrieved May 01, 2017, from http://fortune.com/2017/01/09/anthem-cyber-attack-foreign-government/

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

Rashid, F. Y. (2017, April 27). Annual Verizon security report says sloppiness causes most data breaches. Retrieved May 01, 2017, from http://www.infoworld.com/article/3193028/security/annual-verizon-security-report-says-sloppiness-causes-most-data-breaches.html

Sjouwerman, S. (n.d.). Security Awareness Training Blog. Retrieved May 01, 2017, from https://blog.knowbe4.com/verizon-2016-data-breach-report-phishing-tops-the-list-of-increasing-concerns

Appendix

Parker Lutz Data Mining I Final Project.R

This R file contains well-commented script for executing the main analytics tasks for the Spam Detection task described in this report.

Requirements R version 3.3.3 available here: https://cran.r-project.org/

We recommend using RStudio IDE available here: https://www.rstudio.com/

Working Directory.

Read in files by including complete path name in read.csv() or use setwd() to reset working directory.

Libraries.

Script leverages MASS, caret, and ggplot packages. Open R Script contains commands to install and load all necessary libraries.