STAT 432 Project Proposal: Analysis of the Spam Email Database

Table of Contents

Group Name: Chaoyue the Lucky Fish

Group Member Information

- Bingkun Luo: bluo5
- Xinran Wang: xwang258
- Junchu Zhang: jzhng156

Introduction and Literature Review

Data source information

A Comprehensive Introduction of Data

This dataset is about the categorization of spam emails, which is a problem faced by all of us. The “spam” concept includes advertisements, make money fast schemes, chain letters, pornography, etc.. Spam email is defined by three characteristics, anonymity(unknown sender), mass mailing(large number of recipients) and unsolicited(not requested by recipients) . Not only will dealing with spam emails become a waste of time, but some service providers also spend huge amount of money on spam. It’s necessary for data scientists to develop an efficient classification of spam emails. In this way, we found this data set from the UCI Machine Learning Repository about the classification of spam email. The collection of the spam emails in this data set was from the author’s postmaster and individuals who had filed spam, while the collection of non-spam email was from field work and personal emails. There are some specific words or symbols that are typical in spam email and some typical in non-spam email, while the others does not have a significant direction to spam or non-spam. We will use the machine learning methods we learned this semester to analyze this dataset, find out the words and symbols that influence the classification of spam email the most, and compare the different classification methods.

First, we input the data and change the variable names of each columns for future analysis.

```
##   word_freq_make word_freq_address word_freq_all word_freq_3d
## 1           0.00              0.64          0.64           0
## 2           0.21              0.28          0.50           0
```

```
## 3         0.06            0.00        0.71          0
## 4         0.00            0.00        0.00          0
## 5         0.00            0.00        0.00          0
## 6         0.00            0.00        0.00          0
##   word_freq_our word_freq_over word_freq_remove word_freq_internet
## 1         0.32          0.00             0.00                0.00
## 2         0.14          0.28             0.21                0.07
## 3         1.23          0.19             0.19                0.12
## 4         0.63          0.00             0.31                0.63
## 5         0.63          0.00             0.31                0.63
## 6         1.85          0.00             0.00                1.85
##   word_freq_order word_freq_mail word_freq_receive word_freq_will
## 1            0.00           0.00              0.00           0.64
## 2            0.00           0.94              0.21           0.79
## 3            0.64           0.25              0.38           0.45
## 4            0.31           0.63              0.31           0.31
## 5            0.31           0.63              0.31           0.31
## 6            0.00           0.00              0.00           0.00
##   word_freq_people word_freq_report word_freq_addresses word_freq_free
## 1             0.00             0.00                0.00           0.32
## 2             0.65             0.21                0.14           0.14
## 3             0.12             0.00                1.75           0.06
## 4             0.31             0.00                0.00           0.31
## 5             0.31             0.00                0.00           0.31
## 6             0.00             0.00                0.00           0.00
##   word_freq_business word_freq_email word_freq_you word_freq_credit
## 1               0.00             1.29          1.93              0.00
## 2               0.07             0.28          3.47              0.00
## 3               0.06             1.03          1.36              0.32
## 4               0.00             0.00          3.18              0.00
## 5               0.00             0.00          3.18              0.00
## 6               0.00             0.00          0.00              0.00
##   word_freq_your word_freq_font word_freq_000 word_freq_money word_freq_hp
## 1           0.96              0          0.00             0.00            0
## 2           1.59              0          0.43             0.43            0
## 3           0.51              0          1.16             0.06            0
## 4           0.31              0          0.00             0.00            0
## 5           0.31              0          0.00             0.00            0
## 6           0.00              0          0.00             0.00            0
##   word_freq_hpl word_freq_george word_freq_650 word_freq_lab
## 1             0                0             0             0
## 2             0                0             0             0
## 3             0                0             0             0
## 4             0                0             0             0
## 5             0                0             0             0
## 6             0                0             0             0
##   word_freq_labs word_freq_telnet word_freq_857 word_freq_data
## 1              0                0             0              0
## 2              0                0             0              0
## 3              0                0             0              0
## 4              0                0             0              0
```

```
## 5            0             0          0           0
## 6            0             0          0           0
##   word_freq_415 word_freq_85 word_freq_technology word_freq_1999
## 1            0            0                   0         0.00
## 2            0            0                   0         0.07
## 3            0            0                   0         0.00
## 4            0            0                   0         0.00
## 5            0            0                   0         0.00
## 6            0            0                   0         0.00
##   word_freq_parts word_freq_pm word_freq_direct word_freq_cs
## 1               0            0             0.00            0
## 2               0            0             0.00            0
## 3               0            0             0.06            0
## 4               0            0             0.00            0
## 5               0            0             0.00            0
## 6               0            0             0.00            0
##   word_freq_meeting word_freq_original word_freq_project word_freq_re
## 1                 0              0.00                 0        0.00
## 2                 0              0.00                 0        0.00
## 3                 0              0.12                 0        0.06
## 4                 0              0.00                 0        0.00
## 5                 0              0.00                 0        0.00
## 6                 0              0.00                 0        0.00
##   word_freq_edu word_freq_table word_freq_conference char_freq_;
## 1          0.00               0                    0       0.00
## 2          0.00               0                    0       0.00
## 3          0.06               0                    0       0.01
## 4          0.00               0                    0       0.00
## 5          0.00               0                    0       0.00
## 6          0.00               0                    0       0.00
##   char_freq_(  char_freq_[ char_freq_! char_freq_$ char_freq_#
## 1       0.000          0       0.778       0.000       0.000
## 2       0.132          0       0.372       0.180       0.048
## 3       0.143          0       0.276       0.184       0.010
## 4       0.137          0       0.137       0.000       0.000
## 5       0.135          0       0.135       0.000       0.000
## 6       0.223          0       0.000       0.000       0.000
##   capital_run_length_average capital_run_length_longest
## 1                      3.756                         61
## 2                      5.114                        101
## 3                      9.821                        485
## 4                      3.537                         40
## 5                      3.537                         40
## 6                      3.000                         15
##   capital_run_length_total spam
## 1                      278    1
## 2                     1028    1
## 3                     2259    1
## 4                      191    1
## 5                      191    1
## 6                       54    1
```

We didnt't see any missing values in this dataset. So we continue with the dataset and explain the dependent and the independent variables.

There is one nominal {0,1} class attribute of type spam variable, which also known as outcome in the UCI dataset. The value of "0" means the email is categorized as non-spam email, while "1" denotes the email is considered as spam email.

The other 57 variables in the dataset are classified as independent variables, which are inputs. All independent variables are continuous variables, and are divided into several segments. The first 48 continuous real [0,100] attributes of type word_freq_WORD = percentage of words in the e-mail that match WORD. The calculation is 100 * (number of times the WORD appears in the email) / total number of words in email. A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string. The following 6 continuous real [0,100] attributes of type char_freq_CHAR = percentage of characters in the email that match CHAR. The calculation is 100 * (number of CHAR occurences) / total characters in e-mail. There is 1 continuous real [1,…] attribute of type capital_run_length_average = average length of uninterrupted sequences of capital letters, 1 continuous integer [1,…] attribute of type capital_run_length_longest = length of longest uninterrupted sequence of capital letters, 1 continuous integer [1,…] attribute of type capital_run_length_total = sum of length of uninterrupted sequences of capital letters = total number of capital letters in the email.

Literature Review

We found relevant analyzes of the data set from the following resources:

The research group used the same dataset as we do. They wanted to find the best filter of the spam/non-spam email with the highest accuracy rate. They first classified the full dataset using different methods and found that Random Forest provides the best accuracy rate. Then they used the Best-First Feature selection algorithm to select a subset of the features of the original dataset and did the classification again. They found that tree like classifiers (Random trees and Random Forest) works well in spam email detection and accuracy rate improved incredibly when they first applied feature selection algorithm into the entire process.

Summary Statistics and Data Visualization

We separate our dataset into the training set, “train”, and the testing set, “test”. We choose to split the size of the training set and the testing set by 3:1.

```
library(caTools)
set.seed(5)
sample = sample.split(my_data,SplitRatio = 0.75)
train =subset(my_data,sample ==TRUE)
test =subset(my_data, sample==FALSE)
```

We can take a look at the overall summary statistics of the dataset. Because the dataset has 57 independent variables and 1 dependent variable. For now, we would like to show the summary statistic of the independent variables. Due to the space limit, we will only print out the result for the first input to have a general idea of how the dataset looks like.

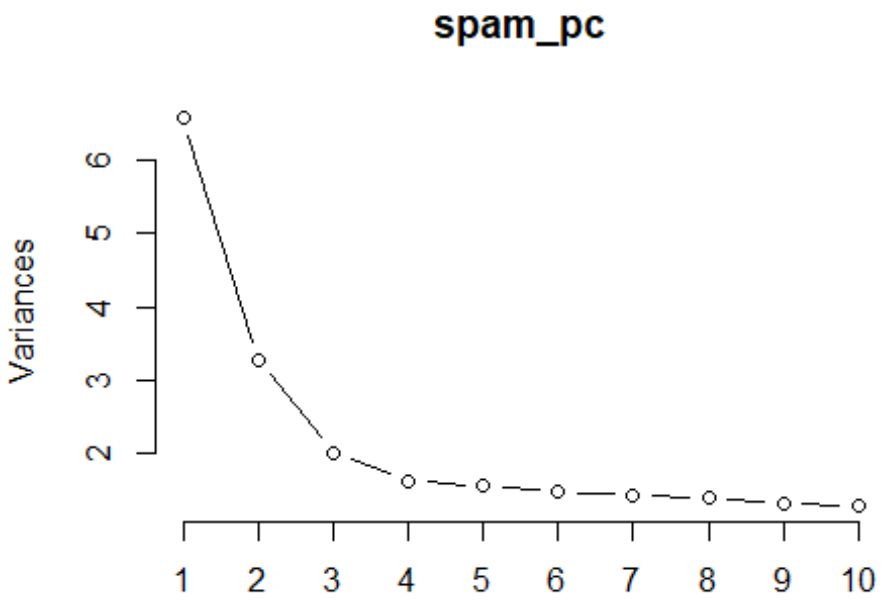
```
## $Mean
## [1] 0.1045534
##
## $Median
## [1] 0
##
## $Max.Min
## [1] 0.00 4.54
##
## $Variance
## [1] 0.09324324
##
## $Std.Dev
## [1] 0.3053576
##
## $Coeff.Variation.Prcnt
## [1] 292.0591
##
## $Std.Error
## [1] 0.004501762
##
## $Quantile
##   0%  25%  50%  75% 100%
## 0.00 0.00 0.00 0.00 4.54
```

After we look into the x variables, our group would also like to examine the dependent variable, spam. This is a binomial factor, in which the value “1” represents spam email, and the value of “0” represents non-spam email.

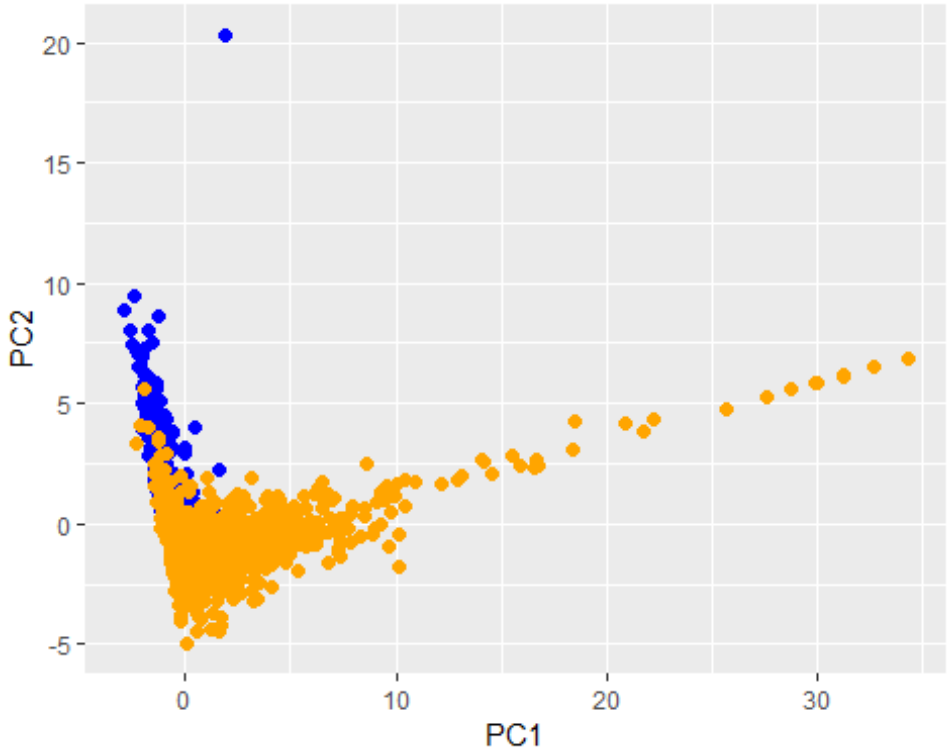
```
## $Mean
## [1] 0.3940448
##
## $Median
## [1] 0
##
## $Max.Min
## [1] 0 1
##
## $Variance
## [1] 0.2388254
##
## $Std.Dev
## [1] 0.4886977
##
## $Coeff.Variation.Prcnt
## [1] 124.0208
##
## $Std.Error
## [1] 0.007204671
##
## $Quantile
##   0%  25%  50%  75% 100%
##    0    0    0    1    1
```

Since the spam email is classified as a binary value. We could directly view that in the original dataset, the percentage of the spam email is 0.394, which is the mean value. The percentage of non-spam email will be 0.606, which can be calculated by 1-0.394.

After we have a general view of both the independent and dependent variables. We would like to have some visualization of the dataset. First of all, we will perform the Principal Component Analysis (PCA). The plot is shown below:



By using the elbow method, we find that the first two principle components explain most of the variances. So, we will use the first two PC directions to plot the variables and have a clear visualization of the dataset.



Although the full linear regression model gives us an R square larger than 0.5, our group chooses not to use it because the dependent variable, which is spam variable is binomial. Therefore, the assumptions of linear regression are violated, and the predicted value is meaningless. Due to the special case of the spam variable, our group decides to use the logistic regression to run the full model, instead of the linear regression.

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##
## Call:
## glm(formula = spam ~ ., family = binomial(link = logit), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4488  -0.1994   0.0000   0.0838   4.7099
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.610e+00  1.773e-01  -9.083  < 2e-16 ***
## word_freq_make          -6.115e-01  3.072e-01  -1.991 0.046509 *
## word_freq_address       -1.628e-01  9.203e-02  -1.769 0.076919 .
## word_freq_all            2.055e-01  1.315e-01   1.563 0.118078
```

```
## word_freq_3d              2.148e+00  1.980e+00   1.085 0.278143
## word_freq_our             5.143e-01  1.242e-01   4.142 3.45e-05 ***
## word_freq_over            9.963e-01  3.077e-01   3.238 0.001205 **
## word_freq_remove          2.453e+00  4.241e-01   5.784 7.28e-09 ***
## `word_freq_internet `     7.000e-01  2.141e-01   3.269 0.001078 **
## word_freq_order           3.166e-01  3.254e-01   0.973 0.330594
## word_freq_mail            6.826e-02  8.130e-02   0.840 0.401112
## word_freq_receive        -3.233e-01  4.025e-01  -0.803 0.421867
## word_freq_will           -2.536e-01  9.650e-02  -2.628 0.008588 **
## word_freq_people         -1.386e-01  2.614e-01  -0.530 0.595962
## word_freq_report          8.513e-02  1.538e-01   0.554 0.579776
## word_freq_addresses       7.282e-01  7.315e-01   0.996 0.319485
## word_freq_free            1.215e+00  1.928e-01   6.302 2.95e-10 ***
## word_freq_business        1.487e+00  3.352e-01   4.436 9.17e-06 ***
## word_freq_email           2.661e-01  1.607e-01   1.656 0.097662 .
## word_freq_you             6.893e-02  4.267e-02   1.615 0.106210
## word_freq_credit          1.577e+00  7.296e-01   2.162 0.030625 *
## word_freq_your            2.495e-01  6.582e-02   3.790 0.000151 ***
## word_freq_font            1.118e-01  1.712e-01   0.653 0.513918
## word_freq_000             2.363e+00  6.510e-01   3.630 0.000284 ***
## word_freq_money           7.124e-01  2.814e-01   2.532 0.011343 *
## word_freq_hp             -1.987e+00  3.463e-01  -5.739 9.50e-09 ***
## word_freq_hpl            -7.022e-01  4.496e-01  -1.562 0.118306
## word_freq_george         -8.876e+00  2.119e+00  -4.189 2.80e-05 ***
## word_freq_650             7.993e-01  3.035e-01   2.634 0.008444 **
## word_freq_lab            -3.604e+00  2.346e+00  -1.536 0.124487
## word_freq_labs           -2.106e-01  3.409e-01  -0.618 0.536627
## word_freq_telnet         -2.027e-01  5.557e-01  -0.365 0.715276
## word_freq_857             1.193e+00  2.879e+00   0.415 0.678457
## word_freq_data           -7.016e-01  3.471e-01  -2.021 0.043274 *
## word_freq_415             1.605e+00  2.007e+00   0.800 0.423895
## word_freq_85             -1.549e+00  9.932e-01  -1.559 0.118943
## word_freq_technology      6.640e-01  3.446e-01   1.927 0.053979 .
## word_freq_1999           -2.662e-01  2.714e-01  -0.981 0.326676
## word_freq_parts          -6.943e-01  4.377e-01  -1.586 0.112701
## word_freq_pm             -1.144e+00  5.308e-01  -2.156 0.031084 *
## word_freq_direct         -3.020e-01  4.750e-01  -0.636 0.524852
## word_freq_cs             -4.011e+01  2.942e+01  -1.363 0.172744
## word_freq_meeting        -2.067e+00  7.025e-01  -2.942 0.003259 **
## word_freq_original       -1.849e+00  1.177e+00  -1.571 0.116231
## word_freq_project        -2.041e+00  6.808e-01  -2.998 0.002721 **
## word_freq_re             -9.891e-01  1.953e-01  -5.066 4.07e-07 ***
## word_freq_edu            -1.436e+00  3.167e-01  -4.533 5.81e-06 ***
## word_freq_table          -2.939e+00  2.133e+00  -1.378 0.168261
## word_freq_conference     -3.912e+00  1.894e+00  -2.066 0.038863 *
## `char_freq_;`            -1.107e+00  4.713e-01  -2.348 0.018858 *
## `char_freq_( `           -2.761e-01  3.494e-01  -0.790 0.429374
## `char_freq_[`            -1.007e+00  1.302e+00  -0.773 0.439478
## `char_freq_!`             2.562e-01  5.982e-02   4.283 1.85e-05 ***
## `char_freq_$`             6.163e+00  9.357e-01   6.586 4.51e-11 ***
## `char_freq_#`             2.035e+00  1.265e+00   1.609 0.107674
```

```
## capital_run_length_average  5.857e-02  4.793e-02   1.222 0.221797
## capital_run_length_longest  1.130e-02  3.400e-03   3.325 0.000883 ***
## capital_run_length_total    6.450e-04  2.534e-04   2.545 0.010919 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4576.1  on 3411  degrees of freedom
## Residual deviance: 1276.4  on 3354  degrees of freedom
## AIC: 1392.4
##
## Number of Fisher Scoring iterations: 14
```

From the summary of binomial regression, we can see that at least some variables are significant due to their p-values are less than 0.05.

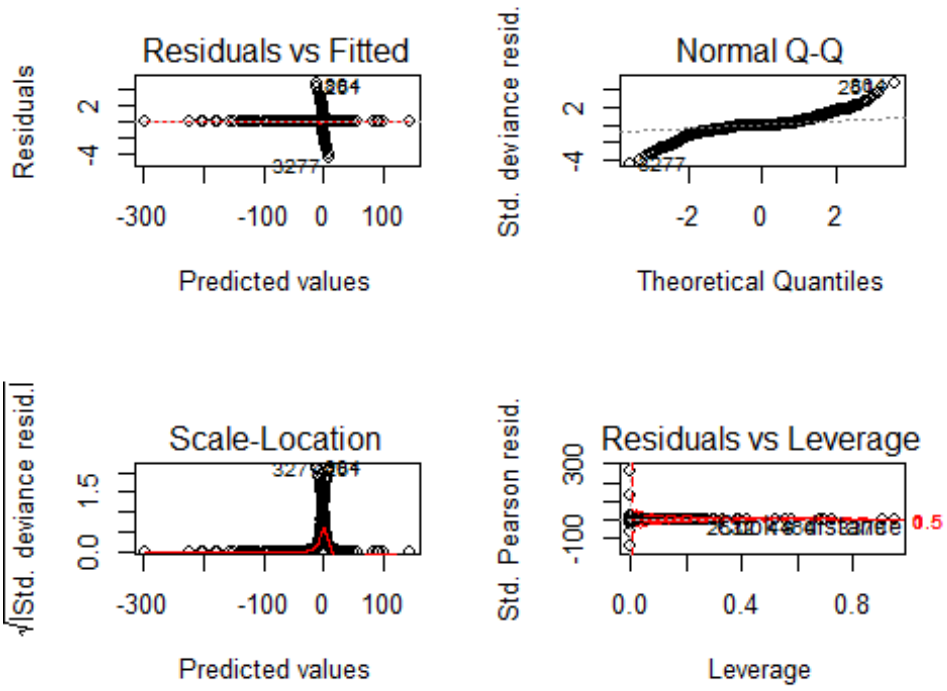
Moreover, we can use the testing set to check how the model is performing and use the confusion matrix to compare the result to the original spam/non-spam classification. In this case, we set 0.5 as the cutoff point. If the predicted value is larger than 0.5, we categorize it into the "spam" category. If the predicted value is smaller than 0.5, we categorize it into the "non-spam" category.

```
##                 actual
## predicted    0       1
##     FALSE  670   47
##     TRUE    51     421
```

After utilizing the testing set, our group also calculate the misclassification error of binomial.

```
## [1] 0.0824222
```

The misclassification error is 0.0824, which implies that in the logistic regression, 91.76% of the testing sample is correctly classified.



We also check the plot of the binomial regression and see that there are no apparent outliers or influential points.

Proposed Analysis

Our group performs the statistics summary for the independent and dependent variable first to have a general view of the dataset. Then our group chooses several methods to plot the dataset and have a data visualization, which have been explained in the previous section.

Our group decides to utilize the testing set to check how the model is performing and use the confusion matrix to compare the result to the original spam/non-spam classification. We first use the binomial regression to fit the independent and dependent variable in the spam dataset due to the reason that the dependent variable is a binary value.

Next, our group will try some other classification methods with the full model and calculate the misclassification error. Since the spam/non-spam column as our "output" variable, all the methods we use should be supervised learning methods. After the classification, our group would

like to compare the misclassification error and the accuracy rate and use it to compare the effectiveness of different models.

1. Naive Bayes

Naive bayes is a binary classifier for the dependent variable, and the words appeared in the email, which are the inputs, is considered as independent. We can use Naive Bayes and calculate the misclassification error and the accuracy rate for this method.

```
## Loading required package: lattice

## Confusion Matrix and Statistics
##
##          Reference
## Prediction   0   1
##          0   388  21
##          1   333 447
##
##                Accuracy : 0.7023
##                95% CI : (0.6754, 0.7282)
##                No Information Rate : 0.6064
##                P-Value [Acc > NIR] : 3.497e-12
##
##                 Kappa : 0.4416
##                Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5381
##             Specificity : 0.9551
##             Pos Pred Value : 0.9487
##             Neg Pred Value : 0.5731
##             Prevalence : 0.6064
##             Detection Rate : 0.3263
##             Detection Prevalence : 0.3440
##              Balanced Accuracy : 0.7466
##
##       'Positive' Class : 0
##
```

From the summary, we could see that the accuracy rate is 0.7023, which implies that the misclassification error is 0.2977, which can be calculates as 1-0.7023. The results show that 70.23% of the testing sample is correctly classified in the Naive Bayes method. Compare these statistics with the one in the logistic regression, we would see that the accuracy rate is much lower for the Naive Bayes method, and we would like to conclude that the Naive Bayes method is not as effectiveness as the logistic regression.

2. SVM

The third method our group would like to try is the SVM method. The aim of SVM aim is to find a best linear function for classification that maximizes the margin. Our group will calculate the misclassification error and the accuracy rate to see whether the SVM method is effective.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0     681  38
##          1     40 430
##
##               Accuracy : 0.9344
##               95% CI : (0.9188, 0.9478)
##     No Information Rate : 0.6064
##     P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.8627
##  Mcnemar's Test P-Value : 0.9099
##
##             Sensitivity : 0.9445
##             Specificity : 0.9188
##             Pos Pred Value : 0.9471
##             Neg Pred Value : 0.9149
##             Prevalence : 0.6064
##             Detection Rate : 0.5728
##             Detection Prevalence : 0.6047
##             Balanced Accuracy : 0.9317
##
##       'Positive' Class : 0
##
```

From the above table, the accuracy rate for the SVM model is 0.9344 and the misclassification error is 0.0656, which is 1-0.9344. This data means that 93.44% of the testing sample is correctly classified in the SVM method. By comparing the results with the Naive Bayes and the logistic regression, we find that the SVM method performs better in classifying the full dataset.

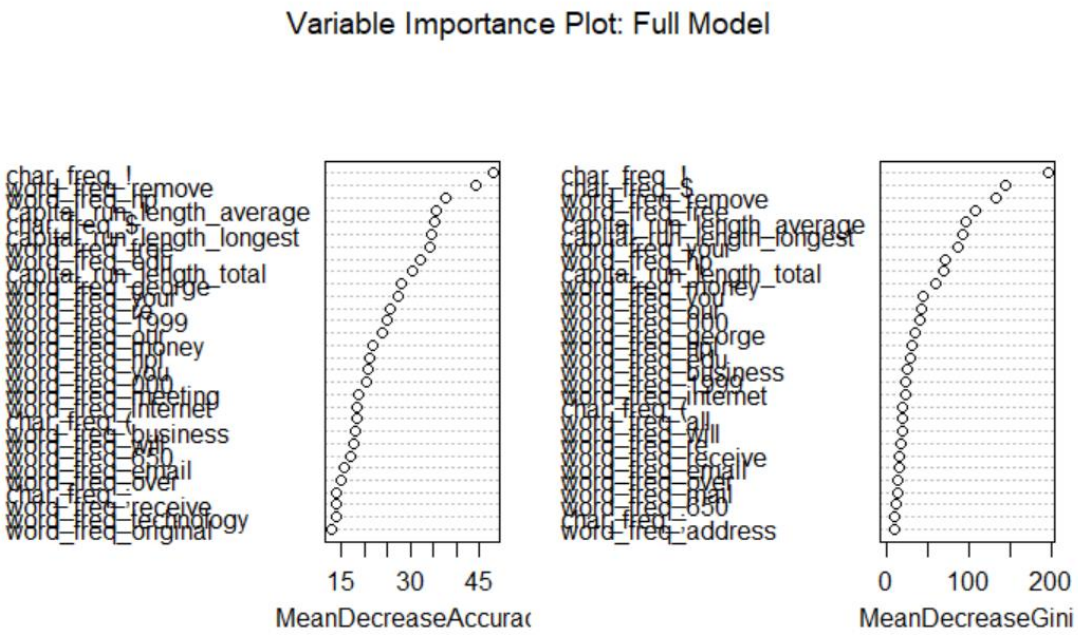
3. Random Forest

The last method we are going to talk about is the Random Forest. The object of the Random Forest is to average multiple deep decision trees, trained on various parts of the same training set. We will increase some bias and loss some interpretability, but we expect to boost our

performance for the final model. Our group will also calculate the misclassification error and the accuracy rate to measure the effectiveness.

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## Confusion Matrix and Statistics
##
##               Reference
## Prediction    0     1
##          0       696   28
##          1        25    440
##
##               Accuracy : 0.9554
##                 95% CI : (0.9421, 0.9664)
##     No Information Rate : 0.6064
##     P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.9065
##  Mcnemar's Test P-Value : 0.7835
##
##             Sensitivity : 0.9653
##             Specificity : 0.9402
##             Pos Pred Value : 0.9613
##             Neg Pred Value : 0.9462
##             Prevalence : 0.6064
##             Detection Rate : 0.5854
##             Detection Prevalence : 0.6089
##             Balanced Accuracy : 0.9527
##
##        'Positive' Class : 0
##
```

By looking at the result from the above table, the accuracy rate for the Random Forest method is 0.9554 and the misclassification rate is 0.0446, which is calculated by 1-0.9554. The outcome indicates that 95.54% of the testing sample is correctly classified in Random Forest method. The Random Forest method generates the best outcome by comparing the data from the previous methods in the full dataset.



We draw the variable importance plot for the Random Forest of the full model. The importance of the variables in classifying are listed from top to buttom in the variance importance plot. We find that "char_freq_!" is the most important variable in classifying based on both the MeanDecreaseAccuracy and the MeanDecreaseGini.

After analyzing the full dataset with different methods. Our group plan to eliminate some variables based on model selection methods because we find that some of the variables are not significant in the logistic regression. So, our group would like to try some reduced models and test the confusion matrix and accuracy rate of the previous models to see whether we could get a better result. First, we would like to use the "stepwised selection" based on minimum AIC values to generate our reduced model and find out the significant variables.

```
summary(reduced)$coefficient
```

```
##                        Estimate   Std. Error   z value
## (Intercept)          -1.676686141 1.683584e-01 -9.959029
## word_freq_make       -0.721117606 2.816389e-01 -2.560433
## word_freq_address    -0.147615837 8.585123e-02 -1.719438
## word_freq_all         0.209580588 1.289846e-01  1.624850
## word_freq_3d          2.093307705 1.932101e+00  1.083436
## word_freq_our         0.522417467 1.246446e-01  4.191257
## word_freq_over        0.940360062 3.038184e-01  3.095138
## word_freq_remove      2.424723263 4.169907e-01  5.814813
```

```
## `word_freq_internet `        0.702544822 2.100548e-01  3.344579
## word_freq_will              -0.253013046 9.583033e-02 -2.640219
## word_freq_free               1.201492780 1.917070e-01  6.267338
## word_freq_business           1.454117187 3.320099e-01  4.379740
## word_freq_email              0.274715165 1.560723e-01  1.760179
## word_freq_you                0.080141563 4.200475e-02  1.907917
## word_freq_credit             1.784521277 6.989628e-01  2.553099
## word_freq_your               0.240644551 6.196174e-02  3.883761
## word_freq_000                2.302409734 6.472129e-01  3.557422
## word_freq_money              0.752936190 2.904121e-01  2.592647
## word_freq_hp                -2.048705055 3.434637e-01 -5.964836
## word_freq_hpl               -0.680931755 4.393972e-01 -1.549695
## word_freq_george            -9.067989344 1.937745e+00 -4.679662
## word_freq_650                0.799909851 3.040329e-01  2.630997
## word_freq_lab               -3.600545222 2.336276e+00 -1.541147
## word_freq_data              -0.734938172 3.518994e-01 -2.088489
## word_freq_85                -1.575333536 9.950705e-01 -1.583138
## word_freq_technology         0.681116520 3.439383e-01  1.980345
## word_freq_parts             -0.673876245 4.305557e-01 -1.565131
## word_freq_pm                -1.168089466 5.390384e-01 -2.166987
## word_freq_cs               -37.247975330 2.514196e+01 -1.481507
## word_freq_meeting           -2.079610000 7.159402e-01 -2.904726
## word_freq_original          -1.919808791 1.195262e+00 -1.606182
## word_freq_project           -2.134119826 6.742896e-01 -3.164990
## word_freq_re                -1.009569690 1.952408e-01 -5.170894
## word_freq_edu               -1.473636983 3.191559e-01 -4.617295
## word_freq_table             -2.948135221 2.144877e+00 -1.374501
## word_freq_conference        -4.159553384 1.909576e+00 -2.178261
## `char_freq_;`               -0.880506834 3.396550e-01 -2.592357
## `char_freq_!`                0.261714609 6.006734e-02  4.357020
## `char_freq_$`                6.351012064 9.297697e-01  6.830737
## `char_freq_#`                2.306605974 1.027310e+00  2.245287
## capital_run_length_average   0.064233748 4.511717e-02  1.423710
## capital_run_length_longest   0.011513911 3.333164e-03  3.454349
## capital_run_length_total     0.000639849 2.471329e-04  2.589089
##                             Pr(>|z|)
## (Intercept)                 2.303016e-23
## word_freq_make              1.045417e-02
## word_freq_address           8.553471e-02
## word_freq_all               1.041945e-01
## word_freq_3d                2.786149e-01
## word_freq_our               2.774132e-05
## word_freq_over              1.967211e-03
## word_freq_remove            6.070165e-09
## `word_freq_internet `       8.240759e-04
## word_freq_will              8.285251e-03
## word_freq_free              3.672730e-10
## word_freq_business          1.188208e-05
## word_freq_email             7.837751e-02
## word_freq_you               5.640197e-02
## word_freq_credit            1.067691e-02
```

```
## word_freq_your            1.028532e-04
## word_freq_000             3.745118e-04
## word_freq_money           9.524034e-03
## word_freq_hp              2.448792e-09
## word_freq_hpl             1.212147e-01
## word_freq_george          2.873484e-06
## word_freq_650             8.513466e-03
## word_freq_lab             1.232809e-01
## word_freq_data            3.675372e-02
## word_freq_85              1.133901e-01
## word_freq_technology      4.766478e-02
## word_freq_parts           1.175521e-01
## word_freq_pm              3.023581e-02
## word_freq_cs              1.384716e-01
## word_freq_meeting         3.675749e-03
## word_freq_original        1.082339e-01
## word_freq_project         1.550883e-03
## word_freq_re              2.329765e-07
## word_freq_edu             3.887753e-06
## word_freq_table           1.692862e-01
## word_freq_conference      2.938664e-02
## `char_freq_;`             9.532088e-03
## `char_freq_!`             1.318453e-05
## `char_freq_$`             8.447943e-12
## `char_freq_#`             2.474972e-02
## capital_run_length_average 1.545306e-01
## capital_run_length_longest 5.516238e-04
## capital_run_length_total   9.623034e-03
```

After running the stepwise function, we find out that there are 44 variables left in this reduced model based on minimum AIC value. For those significant inputs, our group decide to analyze some of the independent variables and how they contribute to the construction of spam filter.

The collection of the spam-emails is from the postmaster or the individuals who have classified the email as spam. From the significant independent variables, our group has selected several words for spam filter and provide the reason for each of the choices.

- Our

According to the generalized linear model, “Our” is considered as a high potential attribute for spam mail, as its p value is 2.77e-05 (near 0). It is a popular word that is detached among the 4601 correspondence, and we classify it as spam words because lots of the commercial like to include this word in the advertisement emails. For example, “Here is our best-seller products”, “Our lowest price of the season on XX” and so on.

- Free

According to the generalized linear model, “Free” is considered as a high potential attribute for spam mail, as its p value is 3.67e-10l. “Free” by its definition “not costing or charging anything” is considered as a advertising gimmick under most situation. For instance, “free trial”, “free sample” and “free shipping”. There is also some special case for it appears to be a delivery of e-journal, including heading like “O.J. Simpson could be set free from Nevada prison Monday”. But such outrageous news is quite rare, we would instead conclude that “free” is a iconic spam word among the 4601 correspondence.

- Money

In the introduction of the Spambase, we acknowledge that make money fast schemes is one of the popular junk mail circulated across the internet. It is an obvious trap aiming at the credulous people, as “money” is often appears in the multilevel marketing email.

- 000 & $

Very alike the word “Money”, “000” and “$” also are related with large amount of financial crime involving gambling and lottery fraud. For example, “…won $1,000,000”. Moreover, in the table above they each has a p value of 3.75e-04 and 8.45e-12.

Next, we would like to offer a list of non-spam words, and the collection of the non-spam emails is from the field work or the personal emails. Our group will give the analysis of why each of the word are contributed to the non-spam filter.

- George

George is a widespread masculine name for both the given name or surname. It is popular as the given name because of the spread veneration of the Christian military saint Saint George. Also, George serves as a surname of Irish, English, Welsh, South Indian Christian, Middle Eastern Christian, French, or Native American origin. If the email contains the word “George”, it is more likely that people send the email to a specific person named “George” for the personal matters. Therefore, our group classify the word “George” as the non-spam word.

- 650

According to the generalized linear model, “650” is considered as a high potential attribute for non-spam mail, as its p value is 8.51e-03. People often refer the word “650” as the area code. In the situation where the email contains “650”, it is very likely that the transmitters want to send the announcement to the receivers only within a certain area. Therefore, we classify “650” as non-spam filter.

- Project

The word “Project” may refer to the individual or group project in the school work or may refer to the certain tasks/programs the business company are undertaking. Since the work “Project” is often viewed as a sequence of tasks to be executed, the marketing department seldom use this work in their advertisement email. So, we would like to use this word as non-spam filter.

- Re

“Re” in the subject line of the email implies the “response” or “reply”. If the email holds “Re”, it is more likely to be a reply email and have a small chance to become the spam email. So, our group place the word “Re” into the non-spam filter. In addition, “Re” has a p value of 2.33e-07, which is very significant.

- Edu

“Edu” is considered as a high potential attribute for non-spam mail as its p value is 3.89e-06, according to the generalized linear model. Only certain type of people could have the email address ends with “edu”, like the college administration office, professors, students, and so on. If people receive the email from the “edu” address, it is probable that the professors or college faculty member would like to make an announcement or send out the newsletter, which contains the useful information for you. Or even your classmates would like to communicate with you for some group projects or course concerns. Therefore, our group labels “edu” as non-spam filter.

```
significant = c(1,2,4,5,6,7,8,9,12,15,16,17,19,20,21,23,24,25,26,27,28,29,33,34,35,36,38,39,41,42,43,44,
45,46,47,48,49,52,53,54,55,56,57,58)
```

From the summary, we re-fine the significant variables based on AIC value, and we create a new dataset for those. Then, we would like to generate a reduced train and test dataset only for the variables that are selected from the stepwise selection.

In this way, we can use the methods we utilized before to do the classification again and compare the result with the previous analysis for each of the methods. We would like to start with Naive Bayes method first.

1. Naive Bayes

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0     1
##          0     398   23
##          1     323  445
##
##               Accuracy : 0.709
##               95% CI : (0.6823, 0.7347)
##     No Information Rate : 0.6064
##     P-Value [Acc > NIR] : 9.686e-14
##
##                  Kappa : 0.452
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.5520
##             Specificity : 0.9509
##             Pos Pred Value : 0.9454
##             Neg Pred Value : 0.5794
##             Prevalence : 0.6064
##             Detection Rate : 0.3347
##           Detection Prevalence : 0.3541
##           Balanced Accuracy : 0.7514
##
##        'Positive' Class : 0
##
```

The summary table indicates that the accuracy rate is 0.709 and the misclassification error is 0.291, which is 1-0.709 for the Naive Bayes method. There is 70.9% of the testing sample is correctly classified in Naive Bayes method. Compared to the accuracy rate of 0.7023 in the full model, there is a slightly increase in accuracy for this reduced model.

2. SVM

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0      1
##          0      680   39
##          1      41     429
##
##               Accuracy : 0.9327
```

```
##                 95% CI : (0.917, 0.9463)
##    No Information Rate : 0.6064
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.8592
##  Mcnemar's Test P-Value : 0.911
##
##             Sensitivity : 0.9431
##             Specificity : 0.9167
##          Pos Pred Value : 0.9458
##          Neg Pred Value : 0.9128
##          Prevalence : 0.6064
##        Detection Rate : 0.5719
##        Detection Prevalence : 0.6047
##        Balanced Accuracy : 0.9299
##
##       'Positive' Class : 0
##
```

The accuracy rate shown in the above table is 0.9327 and the misclassification error is 0.0673, which can be calculated by 1-0.9327 in the SVM method. The results show that 93.27% of the testing sample is correctly classified. Compared to the 0.9344 accuracy of the full model, there is an insignificant decrease in accuracy rate for the SVM method for the reduced model. However, the SVM method performs better in classification than Naive Bayes method in both the full and the reduced model.

3. Random Forest

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction   0    1
##         0     695  28
##         1     26    440
##
##               Accuracy : 0.9546
##                 95% CI : (0.9412, 0.9657)
##    No Information Rate : 0.6064
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.9048
##  Mcnemar's Test P-Value : 0.8918
##
##             Sensitivity : 0.9639
##             Specificity : 0.9402
##          Pos Pred Value : 0.9613
##          Neg Pred Value : 0.9442
##          Prevalence : 0.6064
```

```
##          Detection Rate : 0.5845
##          Detection Prevalence : 0.6081
##          Balanced Accuracy : 0.9521
##
##       'Positive' Class : 0
##
```

By looking at the result from the above table, the accuracy rate for the Random Forest method is 0. 9546 and the misclassification rate is 0. 0454, which is calculates by 1-0. 9546. There is 95.46% of the testing sample is correctly classified in the reduced model of Random Forest method. Compared to the 0.9537 accuracy rate of the full model, there is a slightly increase for the reduced model. We could also conclude that the accuracy rate in Random Forest method performs the best outcome both in full and the reduced model than thee Naive Bayes method and SVM method.

We have found that Random Forest may be a good method for classification of spam/non-spam email. So, we would like to take a close look at the Random Forest method to see whether we could further improve the effectiveness of the prediction. Therefore, our group plans to reduce the variables based on the p-value of the logistic regression, instead of the "stepwise selection" method. First, we will include the variables which the p-value is less than 0.05 in binomial regression, then we will calculate the misclassification error and the accuracy rate to see whether the new reduced model is more effective.

```
significant2 = c(5,6,7,8,16,17,19,21,23,24,25,27,28,33,34,36,42,44,45,46,49,52,53,54,55,56,57,58)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction   0     1
##         0     693   27
##         1     28    441
##
##               Accuracy : 0.9537
##                 95% CI : (0.9402, 0.965)
##    No Information Rate : 0.6064
##    P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.9031
##  Mcnemar's Test P-Value : 1
##
##            Sensitivity : 0.9612
##            Specificity : 0.9423
##         Pos Pred Value : 0.9625
```

```
##          Neg Pred Value : 0.9403
##          Prevalence : 0.6064
##          Detection Rate : 0.5828
##          Detection Prevalence : 0.6056
##          Balanced Accuracy : 0.9517
##
##       'Positive' Class : 0
##
```

From the table above, we could see that the accuracy rate for the second reduced model is 0.9537, and the misclassification error is 0.0462, which can be calculated by 1-0. 9537. The results show that 95.37% of the testing sample is correctly classified. Compared to the 0.9546 accuracy rate of the previous reduced model of the Random Forest method, the accuracy rate of the second reduced model has slightly decreased to 0.9537.

For curious, our group would like to reduce the full model again to the third reduced model. In this time, we will use the variables with p -values less than 0.001 in binomial regression to see is there any improvement of the accuracy rate for the Random Forest method.

```
significant3 = c(5,7,16,21,23,25,27,34,45,46,52,53,58)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction   0    1
##         0    684   45
##         1     37   423
##
##               Accuracy : 0.931
##               95% CI : (0.9151, 0.9448)
##     No Information Rate : 0.6064
##     P-Value [Acc > NIR] : <2e-16
##
##                  Kappa : 0.8551
##  Mcnemar's Test P-Value : 0.4395
##
##             Sensitivity : 0.9487
##             Specificity : 0.9038
##             Pos Pred Value : 0.9383
##             Neg Pred Value : 0.9196
##             Prevalence : 0.6064
##             Detection Rate : 0.5753
##             Detection Prevalence : 0.6131
##             Balanced Accuracy : 0.9263
##
##       'Positive' Class : 0
##
```

From the table above, we could see that the accuracy rate for the third reduced model is 0.931, and the misclassification error is 0.069, which can be calculated by 1-0. 931. The results show that 93.1% of the testing sample is correctly classified. Compared to the 0.9537 accuracy rate of the second reduced model and 0.9546 accuracy rate of the first reduced model of the Random Forest method, the accuracy rate of the third reduced model with variables' p values less than 0.001 has slightly decreased to 0.931. Therefore, we would like to conclude that the reduce the models which based on significance of the variables did not give us a more accurate Random Forest classification.

Conclusion and Discussion

Summarize Scientific Findings

In summary, our group uses one full model and one reduced model from the "stepwise selection" to perform different classification methods such as Naive Bayes, SVM and Random Forest. Based on the accuracy rate and the misclassification rate of each method, we find that in both the full model and the reduced model, Random Forest method has the highest accuracy rate and the lowest misclassification rate than SVM and Naive Bayes. Also, in both models, SVM performs much better than Naive Bayes based on accuracy rate and misclassification rate. The accuracy rate of the Random Forest can reach 95%, which is considerably effective in spam/non-spam classification.

Our group also compares the classification results before and after we reduced the variables. We find that the full model and the reduced model based on "stepwise selection" do not have significant difference in accuracy rate of the Naive Bayes, the SVM and the Random Forest method. Since we do a "stepwise selection" to obtain the reduced model, we use the model with the minimum AIC. So this reduced model is considered better than the full model with larger AIC. We can also use the other model selection criterias such as Mallow's Cp and BIC to do the model selection.

Besides, our group tries to increase the accuracy rate of the Random Forest method. We reduce the model further based on the p-value of the logistic regression. The second reduced model contains all variables with p-value less than 0.05 and the third reduced model contains all

variables with p-value less than 0.001. We find that when we reduce more variables, the accuracy rate of the Random Forest decreases. Based on our analysis, we still consider the first reduced model based on the “stepwise selection” as the best model that gives us the best classification result.

Potential Pitfalls and Improvements

The Spambase has its limitation as the data was collected during 90s, which is not sufficient for analyzing the spam base for nowadays email. Today the genre of email could be specify into Main, Social media, subscribe promotion and Spam. Also email is not only considered as a popular contact tools, but also the essential identification for your account on the website. The detection of spam would need to drop some of the words out of the spam list such as personal pronoun. Though they appear to be significant in our analysis, they all identify as non spam. Instead of adding those interference, we would contain more meaningful non spam words, such as more names of weekdays, which indicates specific schedule. Also phrase could be a better form to detect spam. For instance, the difference between “CS” and “CS go” . Although these two words all contain “CS” word, “CS go” should be identified as promotion rather than the “CS” on the university course listing.

Another pitfall for our project is there are some of the methods which have been mentioned in the literature review, but our group member could not perform the dataset in the same way as the authors did in the their project. Machine learning is a deep topic, and there are much more room for us to learn beyond the course STAT 432. Therefore, our group believes that there is a better way to perform the reduced model with some high-level feature selection algorithm after we gain more knowledge in the machine learning field in the future.