Alison Lanski
EAST
Statistical Learning

Final Classification Project Writeup

As part of Group 1, I worked with Ken, Marisa, and Russ on the "Spambase" dataset, which provides 57 variables based on characteristics of 4601 emails and provides a designation (1, 0) indicating if an email is spam.[†] Our goal was to improve the rate of correct classifications of non-spam emails. Section 1 summarizes findings from exploratory data analysis; Section 2 explains our preprocessing; Section 3 shows our initial models; Section 4 discusses optimization strategies and outcomes; Section 5 provides discussion and concludes. The Appendix provides all figures references in Section 1-4.

Section 1: Exploratory Data Analysis

This dataset was very straightforward. The 4601 emails were all from a single user (with the name "George") and had no NA values. The final variable indicated if the email was spam or not; the preceding 57 variables each represented some characteristic that could be indicative of the type of email. These characteristics fell into two main categories: frequency per 100 words of a particular word or symbol appearing and various metrics related to the use of capital letters. Many variables did not shows strong distinctions between spam and non-spam emails (e.g. Figure 1), but some variables did an obvious preference for one type of message with divergent distributions (e.g. Figure 2). Given the large prevalence of 0 values in the data, the divergent distributions tend to show a much longer or stronger tail in one class over the other.

[†] Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

In general, variables showed a non-normal distribution, with many values piled up at a natural boundary at 0, and long right-tails (e.g. Figure 3; the high number of 0 values obscures that the right tail stretches to 22.5). Within this pattern, nineteen variables had extremely imbalanced distribution, with frequency values of 0 for 90% or more of the observations. When thinking ahead discriminant analysis, data with such little overall variation like this will not process correctly through the model. In fact, the first attempt to run the base models failed because they weren't able to compute QDA with all variables included. For methods like LDA, inconsistent covariance between classes was a larger concern. Overall covariance showed a very large range in values, ranging from close to 0 to an extreme of 350,000, which suggested that scaling the data might be important as a mitigating strategy.

As a whole, the data had 2788 non-spam emails (60.6%) and 1813 spam emails (39.4%). The documentation on the dataset indicted that previous work with classification models had reached a prediction accuracy rate of 93%. Looking at sample scatterplots of 2 variables (e.g. Figure 4) clearly shows that this data is not obviously linearly separable.

Section 2: Pre-processing

The data came in very clean and therefore did not require much pre-processing: no NA values or invalid data. One inconvenience was that the dataset had generic variable names (e.g. V1, V2). Based on the documentation about this dataset, we manually updated the generic names to descriptive titles (word_freq_make, word_freq_address) for easier interpretation of results. We scaled the data in hopes of helping LDA and QDA, so that no single variable would have undue influence on methods and to reduce the impact of widely different covariances. We also produced a reduced version of the dataset which omitted the 19 variables shown unsuitable for

QDA by our exploratory data analysis. For convenience, we also updated the spam-indicator variable to a factor instead of a numeric. At this point, we were ready to proceed with analysis.

Section 3A: Initial Models – Description

Initially, we tried the main classification methods from this semester with default values: random forest (100 trees; default mtry of 7), SVM, LDA, and QDA. KSVM was omitted to reduce computing time. Models were run using a 60/40 train/test split, with 5-fold cross-validation and stratified holdout. Given the often-similar distribution of spam and non-spam observations within each variable, I expected methods like SVM and Random Forest to perform the best. For these same reasons, QDA and LDA were likely to fare more poorly, with LDA having an additional cost potentially imposed due to the covariances. But to cover our bases, it seemed prudent to try all methods the first time through.

We evaluated the models by recording accuracy, sensitivity, and specificity, with "not spam" as the positive class. We chose these metrics because our first goal was to beat the 93% accuracy overall rate, as reported in the data documentation as the best result of previous efforts. Our second goal was to maximize sensitivity: the rate of correct predictions about non-spam emails. This reasoning was based on the fact that it's more frustrating as a user to have real email message go to spam than to have a few extra spam emails appear in an inbox. It was important, however, to not maximize sensitivity at the expense of a large decrease in specificity (spam emails ending up in an inbox); therefore we included that metric as well.

Section 3B: Initial Models – Insights

By examining boxplots of the two most important measures (accuracy: Figure 5 and sensitivity: Figure 6), the default random forest produced the best result, closely followed by SVM. LDA did reasonably well with sensitivity, but relatively poorly with accuracy. QDA was clearly much farther behind on these metric, though it did the best on specificity (Figure 7), followed by random forest and SVM. Given that our data does not seem to be nicely distributed or particularly distinct (e.g. Figure 4), it makes sense that the random forest, which is not forced to consider all variables at the same time, fares better than the others.

The characteristics of the data discovered in EDW which suggested unsuitability for LDA and QDA seemed to come true. A major concern was the non-normality of the data, with lots of values piled up at or near 0 with long tails to the right, and the lack of strong variation in many variables, with high percentages of 0 values. Second, covariance and correlation were a concern. Graphing the covariance and correlation of the full dataset found a "hotspot" with particularly high values (zoomed in correlations: Figure 8). Even after scaling, the two classes of data (spam and not-spam) have somewhat different covariances. When looking at the scaled covariance by classes, the hotspot remains in the real emails (Figure 9), but not in spam messages (Figure 10). With so many variables it's hard to keep track of which is which, but one of the key hotspot variables in the real email plot (34) doesn't show nearly the same extremes (mostly dark blue) in the spam email plot. (Note also that this plot is of correlation, but for scaled data it provides the same information as covariance). These figures together suggest that our data does not have very similar covariance across classes, even after scaling, which is therefore explains the relatively poor performance of LDA. Where LDA fails, QDA often does better, but is only rewarded on the specificity metric, perhaps because QDA, too, is limited by the non-normal distribution of variables; a more pervasive and potentially important issue than covariance.

Conversely, random forests typically handle non-normal data or correlated data reasonably well. It also makes sense that SVM performed generally better than LDA and QDA: the ability of SVM to work with the soft boundaries provided by the margin means that it should handle our very overlapped-data more easily than LDA. The results of the metric-boxplots overall suggest what we could have already known from EDA: the shape of this data is poorly suited to Discriminant Analyses overall.

Further early tests included a second round of base models with unscaled data. This run resulted in nearly identical boxplots, which suggested that scaling was, in fact, unnecessary since it didn't improve the outcomes of any models. A final test checked for a difference in random forest accuracy between 100 and 200 trees; although 200 trees did perform slightly better, the difference was so small that we judged it not worth the significantly increased computing time.

Section 4A: Model Optimization – Model Selection

Because accuracy and sensitivity are the two most important metrics for us, random forest stood out as the clear winner. Specificity was more of a confirmatory metric, so the surprising success of QDA here did not change our overall impression that random forest was likely our best method. Because SVM wasn't far behind random forest on all metrics, we decided it would be worth trying to optimize as well, given the strong influence of particular values of C or nu on the model.

Section 4B: Model Optimization – Methods to Optimize

Having selected random forest, c-SVM and nu-SVM to optimize, we next ran a second set of runs (50 times through with 5-fold cross-validation) and the following tuning options:

- Random forest mtry from 4 to 10
- Random forest weights of 1/1, 1/0.8, 1/0.5 *reducing weights for spam messages*
- c-SVM: 25 C values ranging from 2^-7 to 2^7
- nu-SVM: 25 nu values ranging from 0.01 to 0.78

The various boxplot outputs from this round of optimization revealed several things. As weighting was added to the random forest, it did slightly worse on accuracy, much worse on specificity, and better on sensitivity (Figures 11-13). Overall, the decrease in specificity was greater than the gain in sensitivity. Returning to our first principles, we decided that if the decrease in specificity was enough to pull down overall accuracy, it was too large to consider acceptable. Based on these results, we decided that an unweighted random forest was the best type of model. To verify, I ran a new random forest loop with mtry values ranging from 4-8 and more intermediate weights (1,1), (1, 0.95), (1. 0.9), (1, 0.85); the results of these trees did not show significant improvement over the unweighted version. In fact, the boxplots show again that on runs with higher sensitivity, we also experience lower specificity, leading to similar overall accuracy.

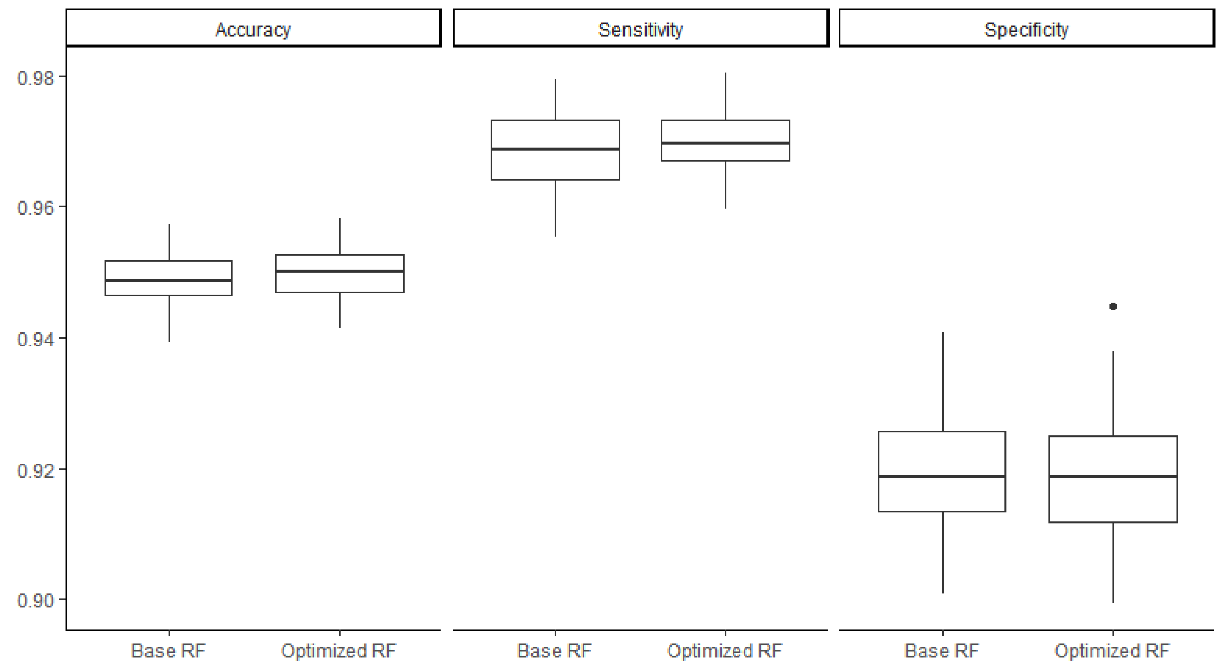
SVM did not fare as well overall on our metrics as the random forest did. The best values of c-SVM occurred at c = 5.34 by all metrics, but all boxplots of c-SVM fell notably below the boxplots for all of the random forest metrics. nu-SM was less consistent about the best value for nu, with nu = 0.17 as the peak of accuracy and specificity, but nu = 0.78 as the peak of sensitivity. These nu values are very far apart, and the boxplots (see Figures 14 and 15; note the extreme difference in y-scale) clearly show that we have an even more severe tradeoff between sensitivity and specificity with nu-SVM; while the specificity at the right edge of nu is very good and can beat a random forest, the sensitivity on that edge is not great. No matter the value of nu, overall accuracy is below what the random forests can offer.

After this stage, we ran one more round of optimization. Here, we were looking to verify the best value of mtry for our random forest, and to compare it to possibly better values of c-SVM. Because the first value of c-SVM and the second value (our best) of c-SVM were separated by a large jump in values, followed by a decline, it seemed reasonable to think that there could be even better c-SVM results in that interval. So this loop used 31 values of c between 0.1 and 6.1. The random forest mtry values in the loop ranged from 1 to 15.

Final boxplots showed that mtry of 6 beat out all other values on accuracy and sensitivity. Using an mtry of 4 had a higher median on specificity, but because this metric was of less interest to us, it did not change our decision. The new values of c-SVM could still not beat random forest. For fun, we did a final optimization trying a variety of kernels with KVM then tuning their parameters; these too failed to beat our random forest model when examining the key metrics.

Section 5: Discussion

Based on the consistent performance of random forest classification, it is the clear winner over SVM as the best of our optimized models. A final loop with two random forests, one using the base parameters and the other using the optimized parameters, shows how narrowly the optimized version beats the base model on accuracy and sensitivity, coming in close to even on specificity.



We also looked at each of the 50 runs of the final loop to determine on which loop the optimized model did best, compared to the base model on that loop (run 43), and on which loop the optimized model did worst, compared to the base model (run 12), as shown in Tables 1 and 2. Confusion matrices with outputs from these runs show that the optimized model can outperform the base model well, and at its worst relative performance only omits 1 "real" email more than the base model. (Specificity is a bit worse, with 10 more spam messages misidentified).

Table 1: Best performance of the optimized model vs. the base model

Base Model Run 43		Actual	
		Not Spam	Spam
Predicted	Not Spam	1065	46
	Spam	50	679

Optimized Model Run 43		Actual	
		Not Spam	Spam
Predicted	Not Spam	1078	46
	Spam	37	679

Table 2: Worst performance of the optimized model vs the base model

Base Model Run 12		Actual	
		Not Spam	Spam
Predicted	Not Spam	1078	60
	Spam	37	665

Optimized Model Run 12		Actual	
		Not Spam	Spam
Predicted	Not Spam	1077	70
	Spam	38	655

Curiously, the best (relative) run of the optimized model and the best (relative) run of the base model have the same specificity, though the optimized model has higher accuracy. These plots and confusion matrices shows that the performance of the optimized forest is only very slightly better after optimization.

In sum, the random forests (regardless of optimization) surpassed not only c-SVM, nu-SVM, KSVM, LDA, and QDA consistently, they also consistently surpassed the accuracy of the null model (60.4% accuracy and sensitivity, with 0% specificity) and the accuracy of previous analysis on this dataset (93%, according to the documentation). This result matched our intuition: with such messy, non-normal data, the method that was best able to distinguish it into pieces, instead of splitting it once as a whole, was likely to do best. Therefore we feel confident in recommending random forest as the best outcome, with 100 trees (or more), suggested mtry of 6, and no weight, with a goal of overall accuracy in the classification of email and a particular emphasis on correct recognition of real (non-spam) messages.

Appendix

Figure 1

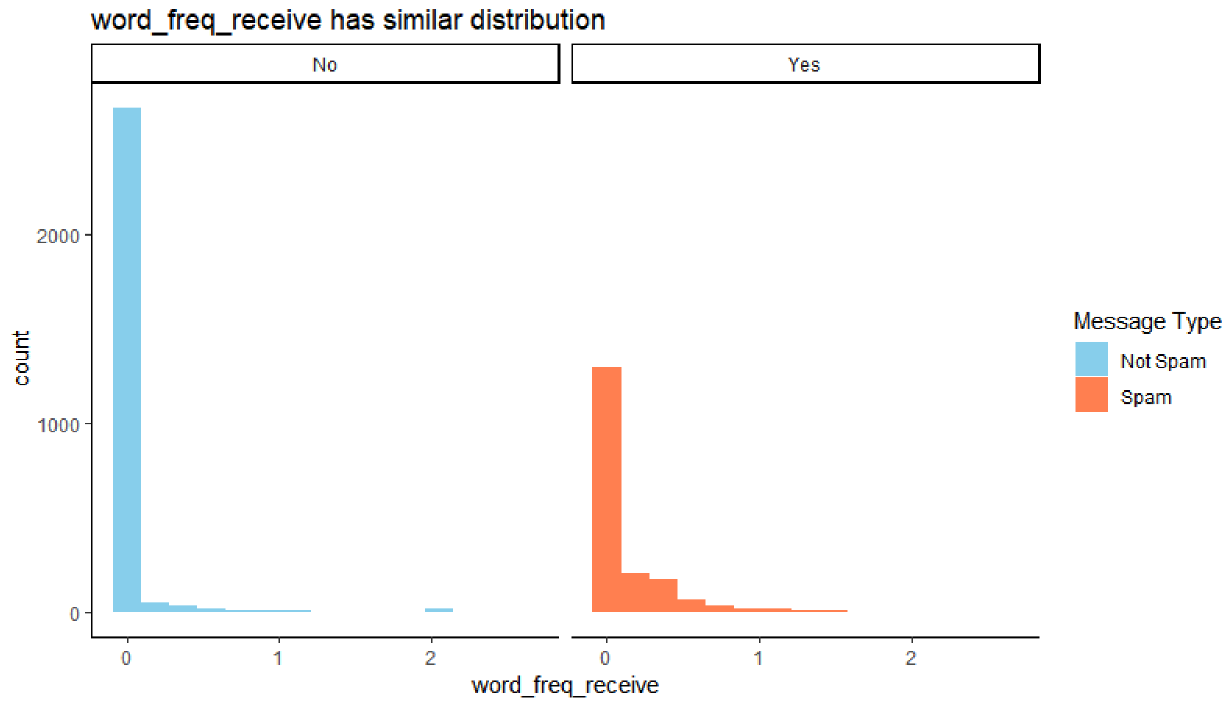


Figure 2

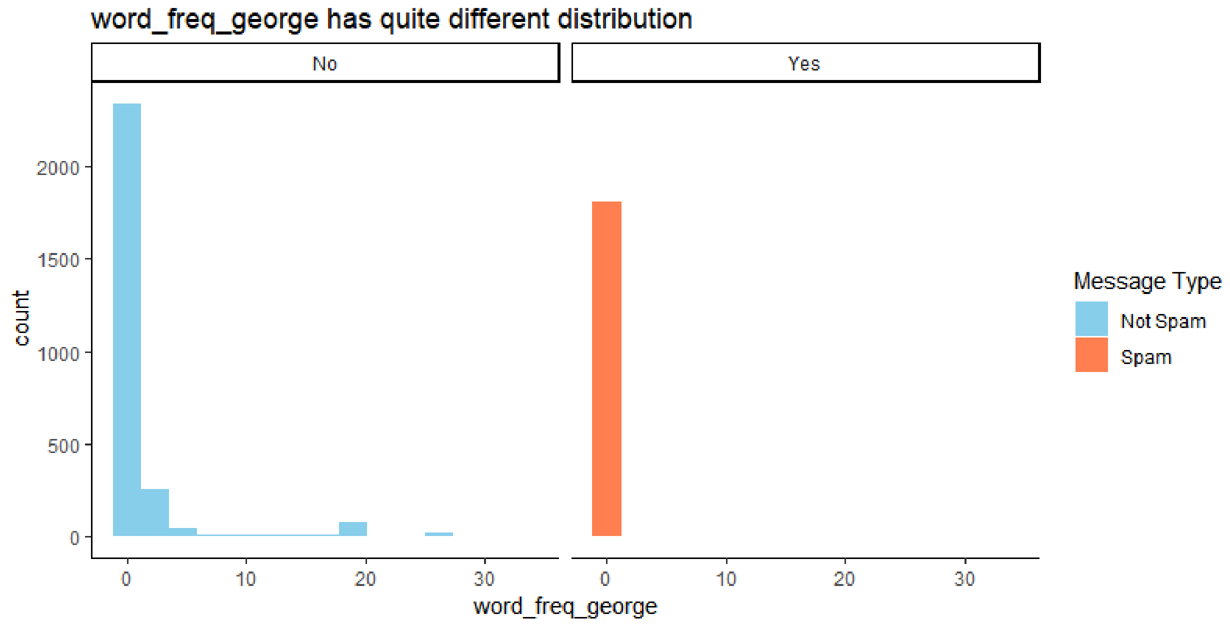


Figure 3

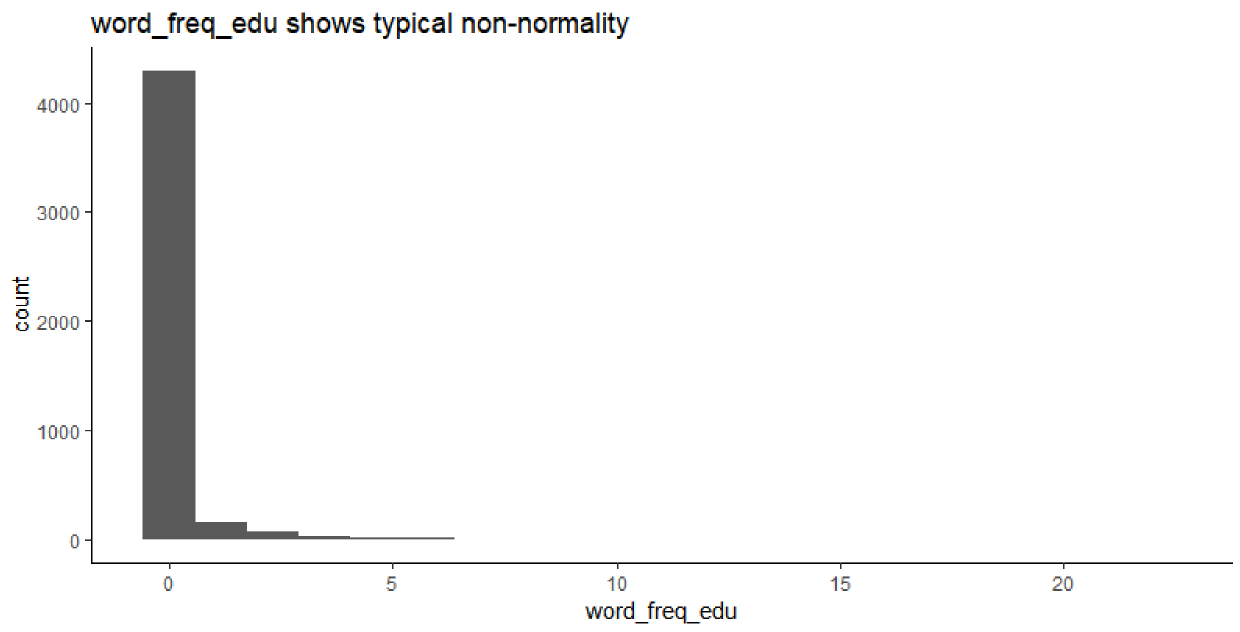


Figure 4

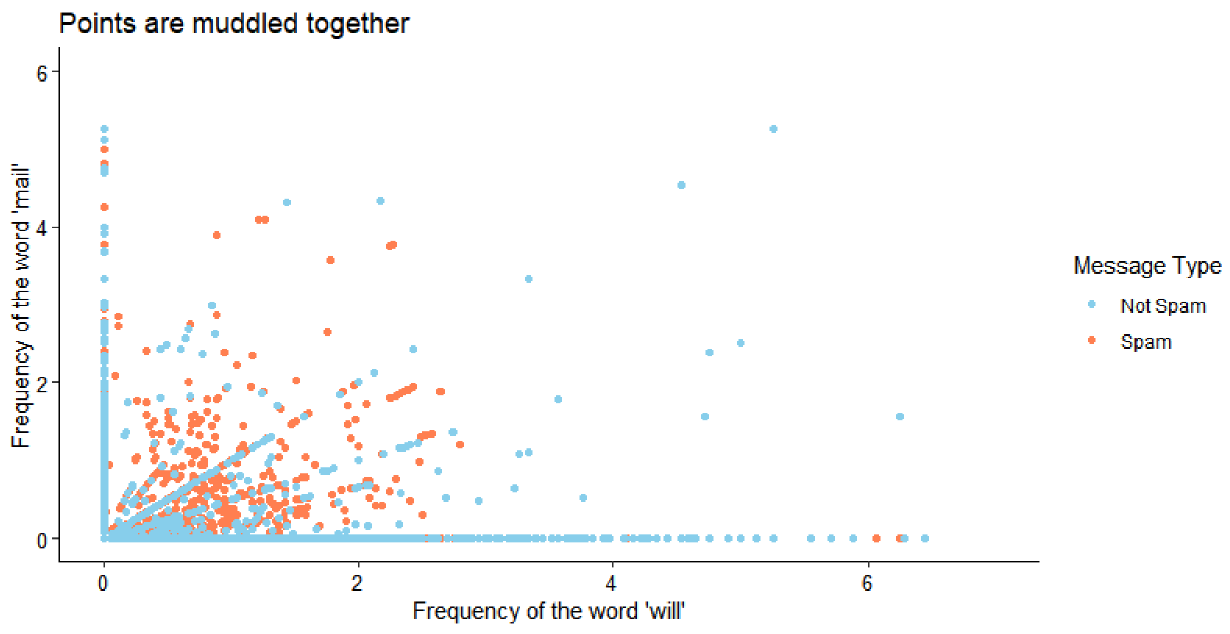


Figure 5

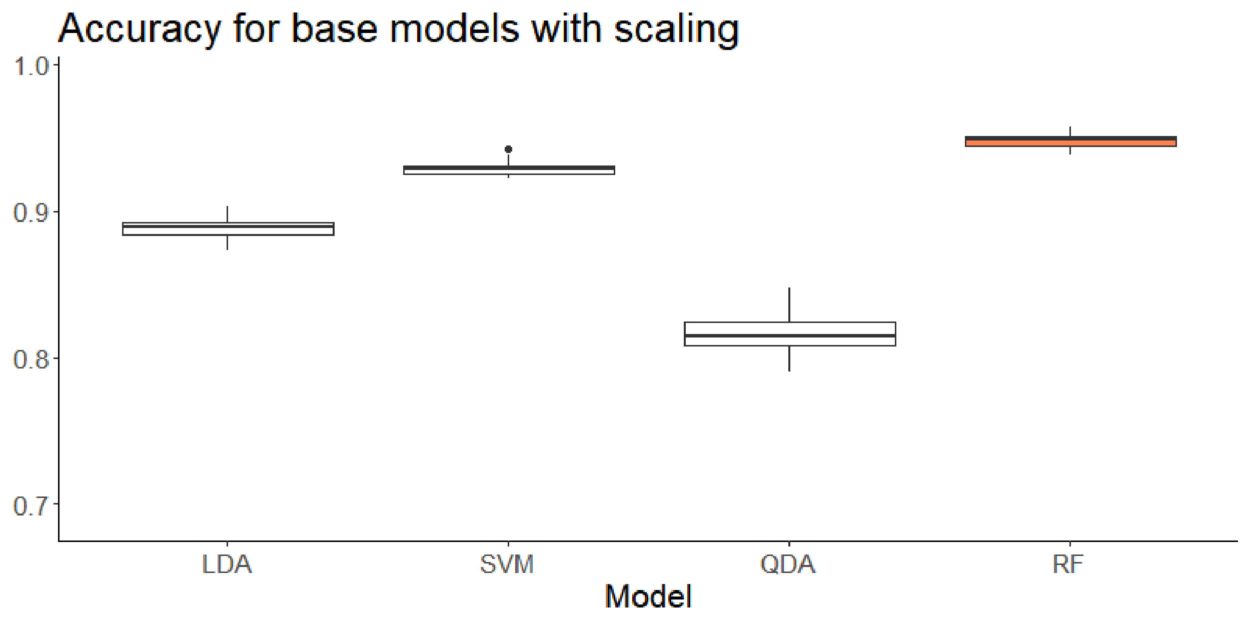


Figure 6

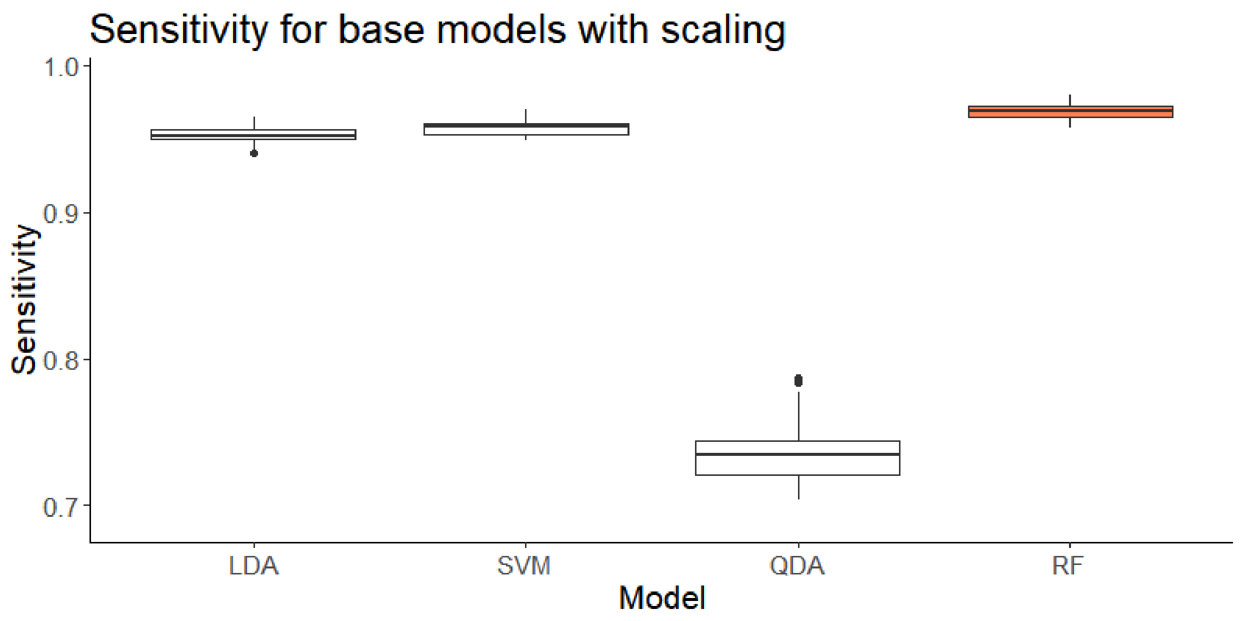


Figure 7

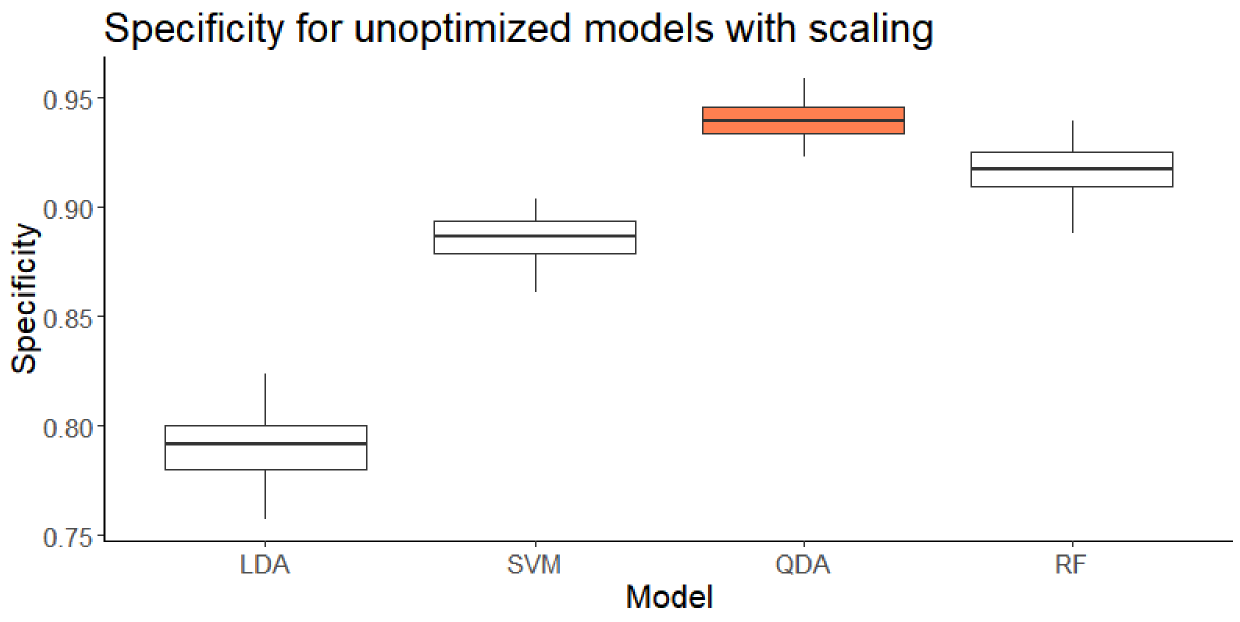


Figure 8

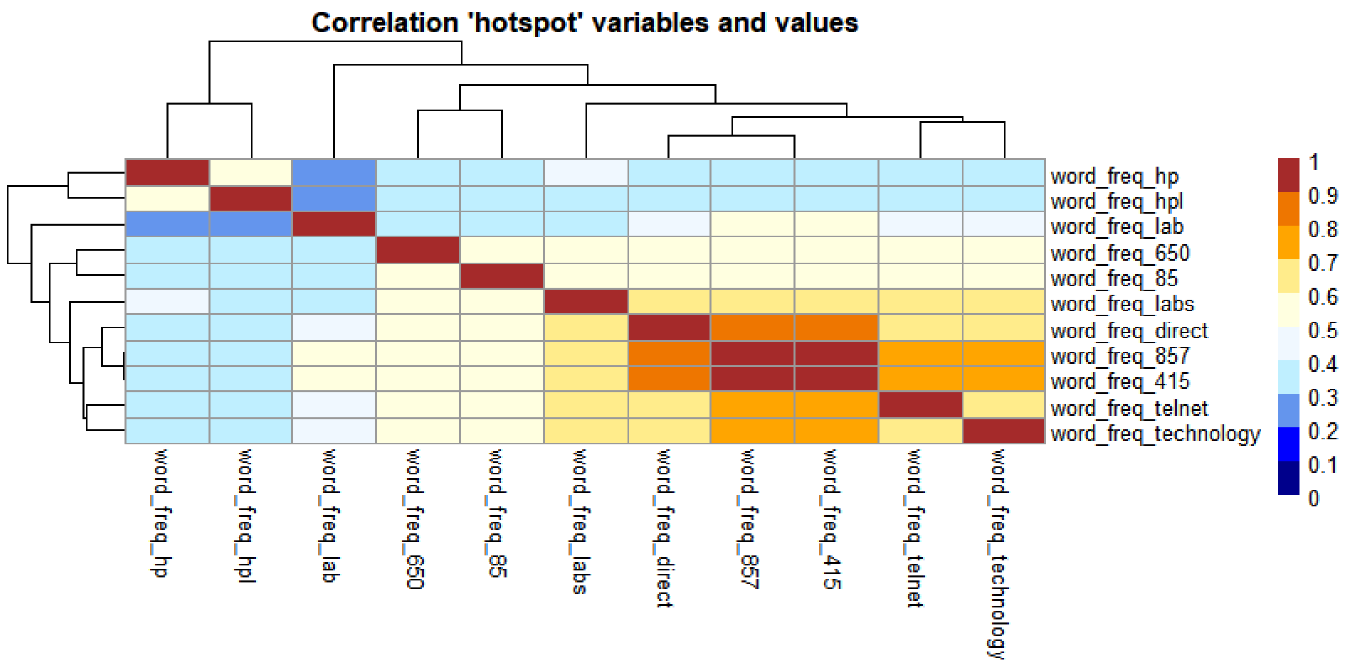


Figure 9

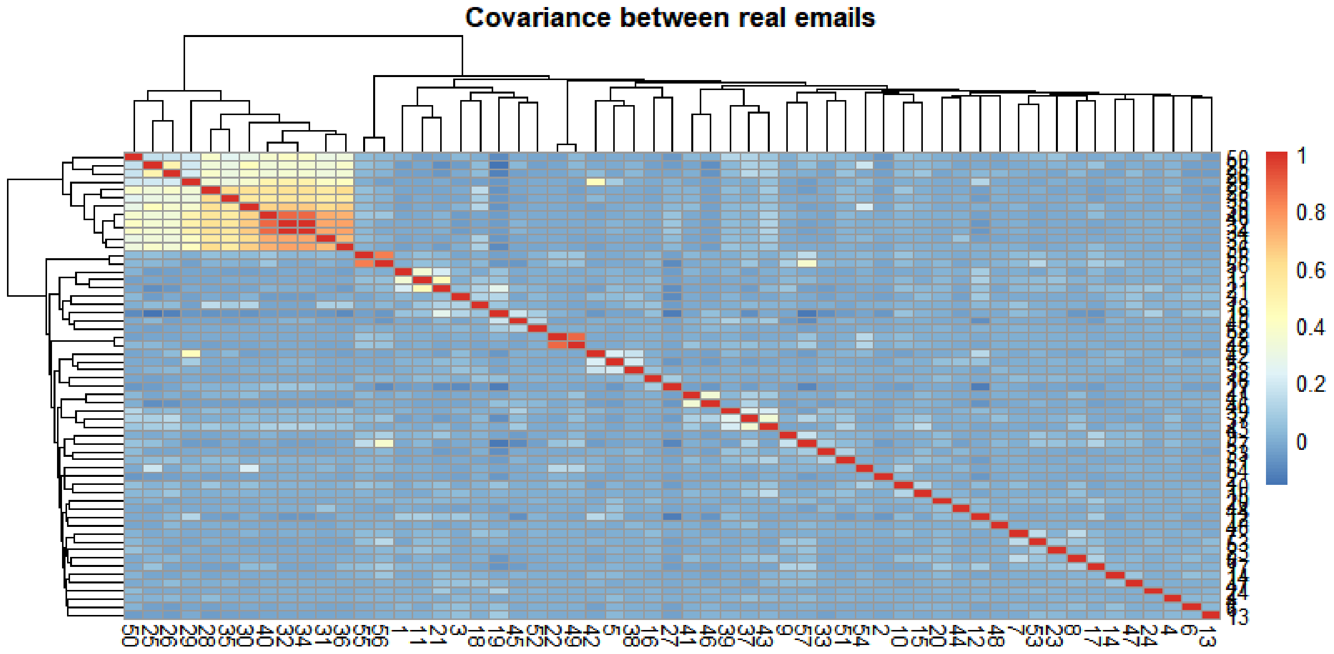


Figure 10

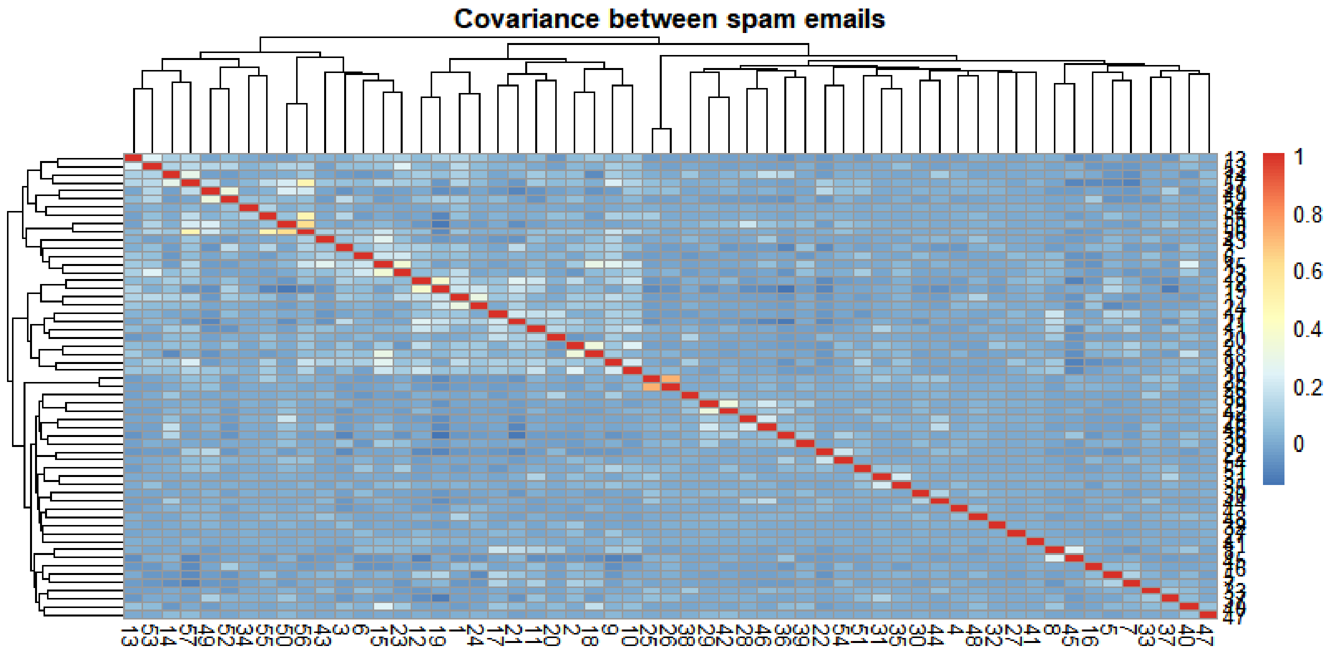


Figure 11

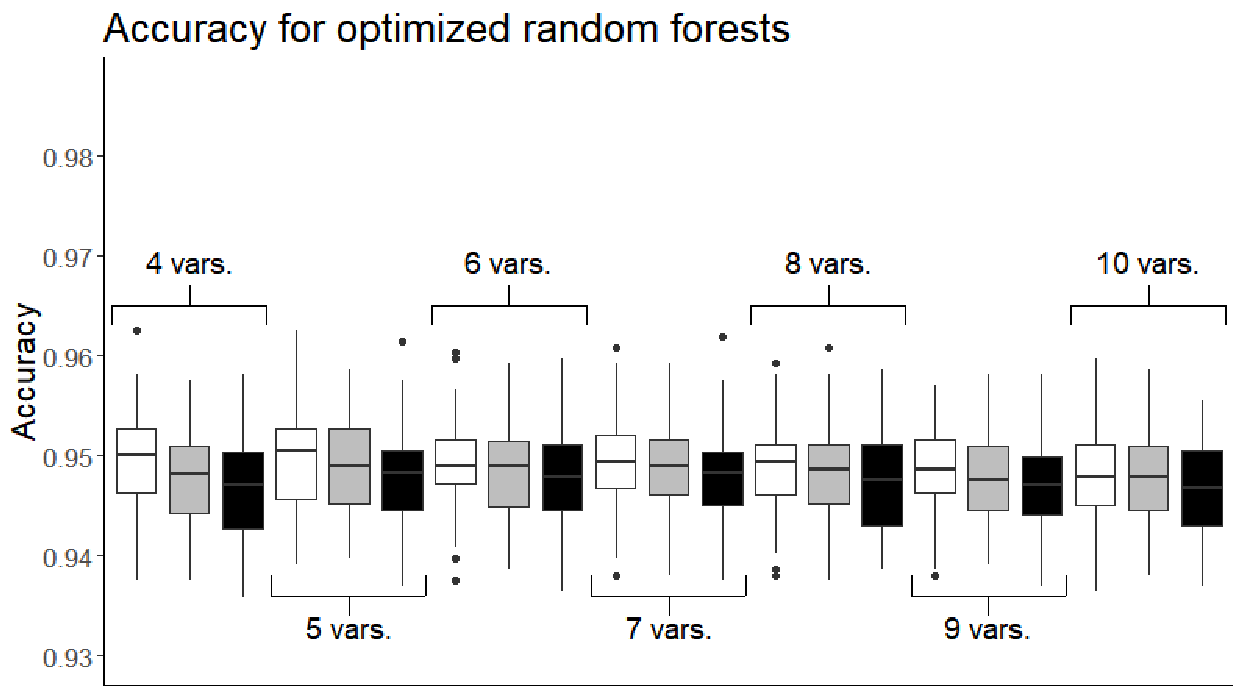


Figure 12

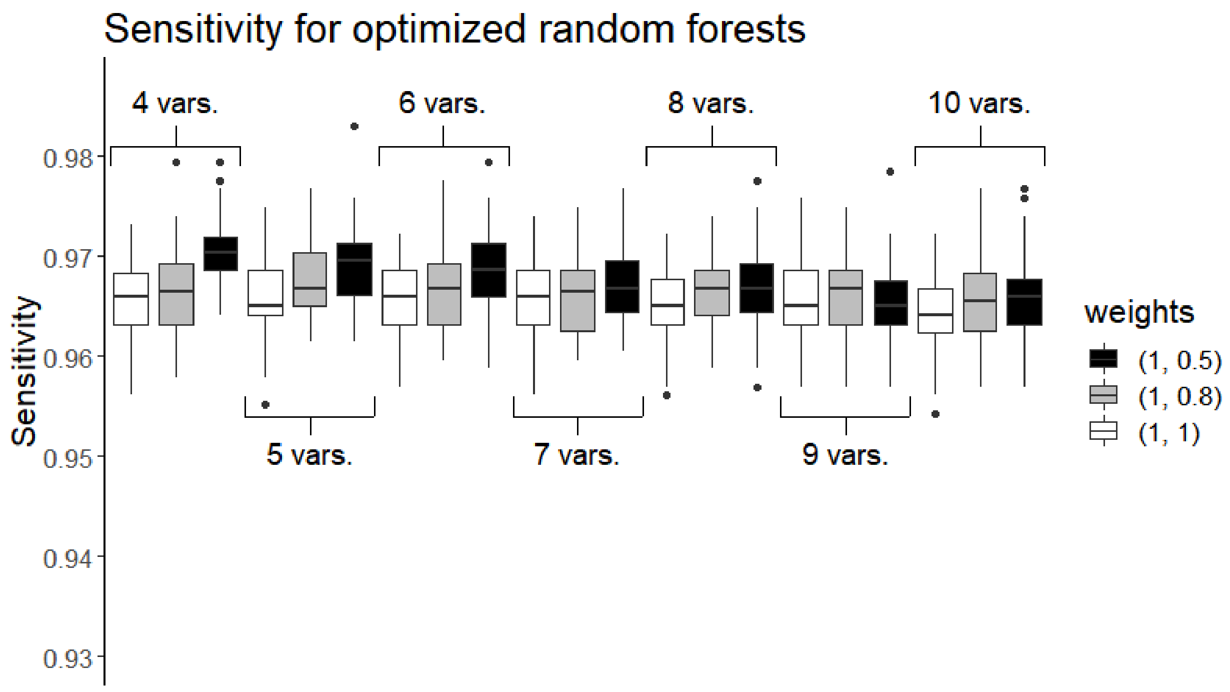
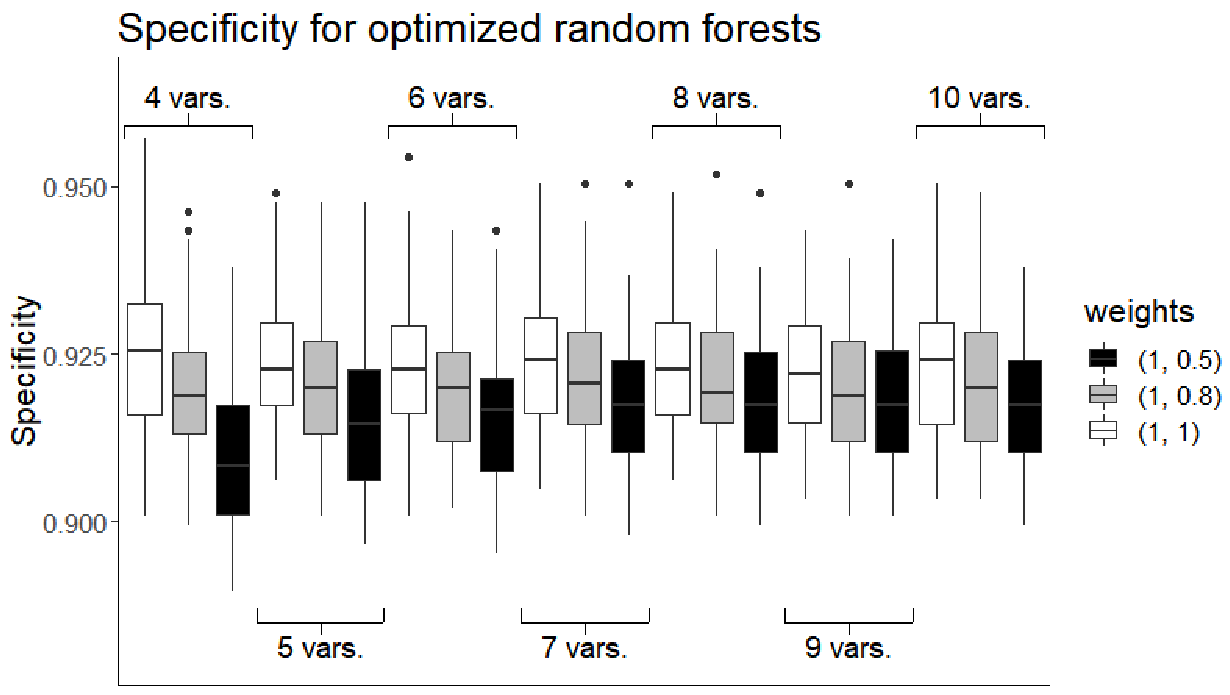


Figure 13



(Figure 14 on the next page, for better comparison of results with Figure 15)

Figure 14

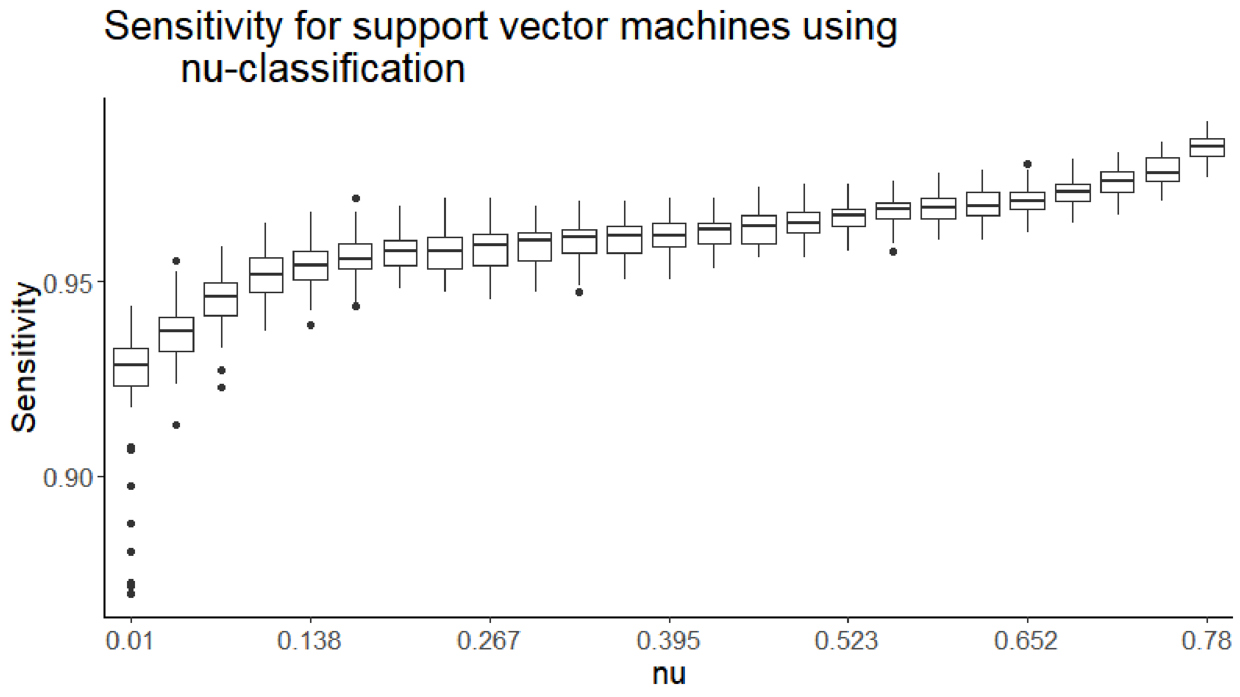
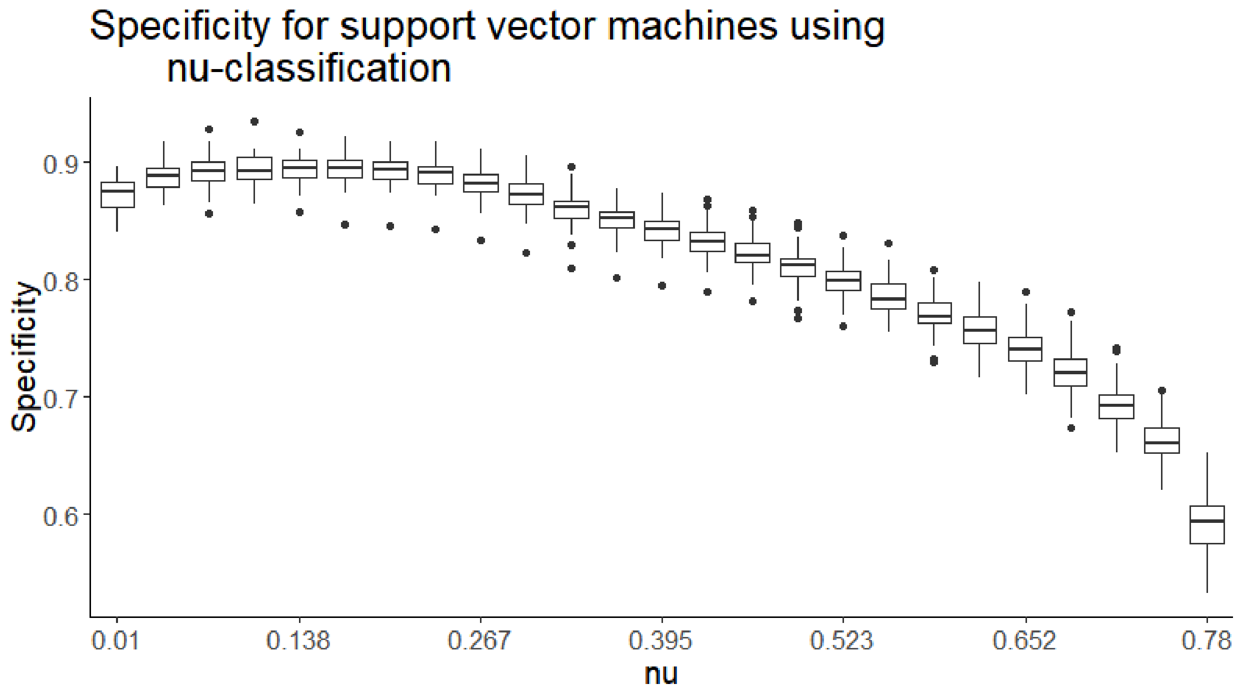


Figure 15


Specificity for support vector machines using nu-classification
Specificity
nu