Draft version December 8, 2016
Preprint typeset using LATEX style AASTeX6 v. 1.0

AN ANALYSIS OF BINARY CLASSIFICATION METHODS VIA SIMULATION AND APPLICATION

Brandon Bergerud, Jeff Hajewski, Sam Justice, Tyler Olson, and Alex Zajichek

ABSTRACT

A plethora of statistical learning methods have arisen over the years to better model the underlying relationships in data. Yet, when performing binary classification, logistic regression is often the first tool of choice among statisticians. In this paper we examine the performance of several machine learning algorithms on various datasets with a binomial response, using both simulated data and real world data, and compare the performance to logistic regression. Overall, we find that no single method outperforms all the others, but recommend regularized logistic regression as an improvement to logistic regression when the number of predictors is comparable to, or exceeds, the number of observations.

Keywords: Binary Classification — Logistic Regression — Neural Networks — Random Forest — Regularized Logistic Regression — Support Vector Machines

1. INTRODUCTION

The purpose of this study is to observe the behavior of classification methods in a controlled binomial environment. Statisticians instinctively reach for logistic regression when presented with a binary response, regardless of other underlying data characteristics. Offering a comparison of binomial classification methods under controlled conditions should sharpen this initial intuition. Logistic regression models, decision trees, neural networks, regularized logistic regression models, and support vector machines will all be fit and tested using sets of simulated data. Properties such as flexibility, robustness, and scalability will be considered along with the traditional sensitivity and specificity metrics to evaluate model performance. Sets of continuous features will be generated from common distributions, and the relationship between the predictors and the response will be examined. Binomial responses are prevalent throughout society, so the simulated data will mimic situations with a binary response.

2. SIMULATION STUDY

Each of the five methods that will undergo investigation involve some sort of fine tuning to obtain an optimal fit. The probability threshold for logistic regression will need to be adjusted if a more conservative or anti-conservative threshold is necessary. Random forests, an ensemble-based decision tree method, require two parameters that involve a minimal amount of optimal specification. The convergence of the out-of-bag (OOB) error, the mean of the training error, will signal the appropriate number of trees. Identifying the number of layers and tuning the adaptive weights of the edges are just two of the primary challenges involved with the implementation of an artificial neural network. An appropriate ridge tuning parameter for regularized logistic regression will be identified using cross validation. Lastly, the optimization of the cost parameter for support vector machines (SVMs) will be induced through a heuristic proposed by Joachims that tunes the inverse of a regularization constant based on the steepest feasible descent.

2.1. *Overview*

A simulation study will be carried out under a plethora of controlled settings. By adjusting the levels of dimensionality and sample size (p and n), the general performance of the selected binomial classification methods will be assessed. For each possible combination of p and n, where $p \in \{1, 10, 50, 100, 500, 1000\}$ and $n \in \{10, 50, 100, 500, 1000, 10000\}$, the test error will be considered and comparisons will be made with the Bayes error rate. In order to allow for a more realistic computation time, only a subset of the possible values of p will be used during the fitting process. Instead of fitting each model with 1 through 1000 predictors for every combination of n and p, appropriate ranges were calculated for each p being examined. The following subsets of predictors will be used to mimic underfitting and overfitting:

- $p = 1$, $range = (1, 2, 3, 4,..., 21)$
- $p = 10$, $range = (1, 2, 4, 6,..., 40)$
- $p = 50$, $range = (10, 14, 18, 22,..., 90)$
- $p = 100$, $range = (20, 28, 36, 44,..., 180)$
- $p = 500$, $range = (340, 356, 372, 388,..., 660)$

- $p = 1000$, *range* = (360, 392, 424, 456,..., 1000)

The baseline setting will impose the standard assumptions of independent, continuous predictors and a true underlying linear relationship between the predictors and response. A future analysis of model performance could introduce characeristics such as multicollinearity, categorical features, and true non-linear relationships, but these will not be tested during this study.

Once the test errors have been obtained, each specified setting will be represented by a plot, which will provide side-by-side comparisons of the errors from each of the five binomial classification methods. Therefore, 36 plots will be constructed to summarize the initial results from this simulation study.

2.2. *Traditional and Regularized Logistic Regression*

Logistic regression was implemented, with observations whose predicted probabilities were higher than 0.5 being assigned to class 1, and observations whose predicted probabilities were less than or equal to 0.5 being assigned to class 0. Regularized logistic regression was also carried out using the ridge regression method. The optimal tuning parameter λ for each model was found via cross-validation.

2.3. *Support Vector Machines*

Using a set of training data where each instance is either identified as being a ”success” or ”failure,” a support vector machine (SVM) constructs a model that assigns new observations to one of the two categories. The SVM creates a hyperplane, and each training data instance is represented as a point in the space. The functional margin, the distance to the nearest training instance in either group, should be as large as possible. When the functional margin is maximized, a clear gap will ideally exist between the two categories, which will in turn minimize the generalization error of the SVM. Each new observation is mapped onto the hyperplane, and its location with respect to the functional margin determines whether the test instance is assigned to class 1 or 0.

For the purpose of this simulation study, the SVM will be formulated as a linear classification method by solving the following optimization problem with respect to α_i:

$$f(x) = \sum_{i}^{N} \alpha_i y_i (x_i^T x) + b \tag{1}$$

2.4. *Neural Networks*

Neural networks consist of a collection of layered nodes (i.e., *neurons*) that take input from one or more nodes from the previous layer and combine with other nodes in the respective layer, along with some activation function which depends on the node input values and node weights. Neural networks, and in particular deep neural networks, are incredibly powerful tools used in achieving state of the art results in various areas of applications from image classification (Krizhevsky et al. 2012) to playing the game Go (Silver et al. 2016). However, given the number of parameters and data points considered in the setting of this experiment, only neural networks with one hidden layer were considered. In fact, several tests were performed with deep networks, for the sake of completeness, and the results were worse than the single hidden layer models by around ten percentage points of test error. Further, they required more time to train and due to their increased flexibility, likely resulted in highly overfitted models (hence the worse tests error).

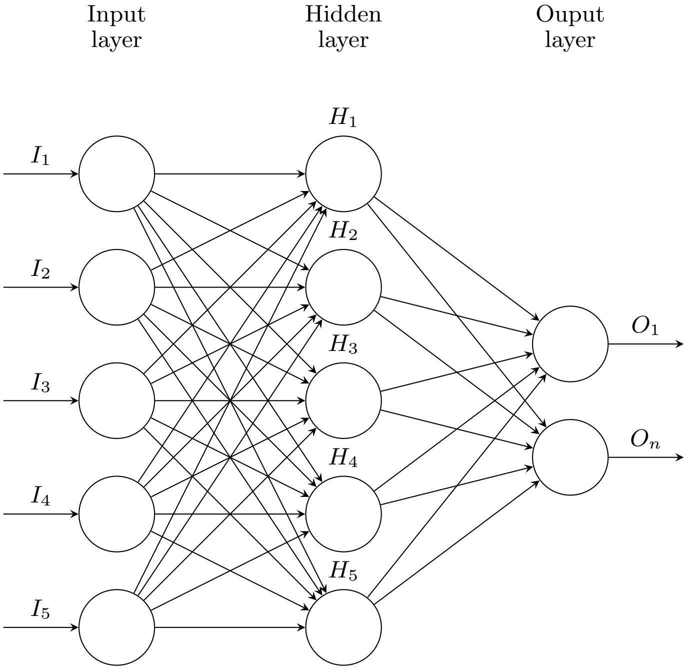

Figure 1. Example neural network for $p = 5$.

A. *Network Architecture*

The networks were built such that the hidden layer had the same number of nodes as the input layer, and the output layer had two nodes, corresponding to 1 and 0. The input node had as many inputs as there were features. Of course, it follows from this network definition that the network architecture changed across the varying p values of the given test and training set, adapting to the size of the problem. See figure 1 for a sample network architecture for $p = 5$.

The input layer uses the rectifier (2) as its activation function while the hidden layer uses the softmax activation function (3).

$$f(x) = \max\{0, f(x)\} \tag{2}$$

$$f(x) = \log(1 + e^x) \tag{3}$$

Where x is actually the inner product of the input vector $\mathbf{x}$ and the weight vector θ.

B. *Training*

Optimization was performed using stochastic gradient descent with momentum as well as Nesterov acceleration (Nesterov 1983). Thus at each step the parameter is updated via equations 4.

$$\begin{aligned} v_{n+1} &= \gamma v_n + \eta \nabla_\theta J(\theta - \gamma v_n) \\ \theta_{n+1} &= \theta_n - \eta_{n+1} \end{aligned} \tag{4}$$

Where θ is the weight vector, γ the learning rate, η the momentum factor, $J(\mathbf{x})$ the cost function, and v_i the momentum term. Stochastic gradient descent was chosen due to the lower computational burden when compared to higher order methods such as BFGS.

Training was somewhat challenging due to the time complexity of the training process. The model training was performed on a 3.4GHz Intel i7 hexacore processor with 64GB of RAM, and a GTX 1080 graphics card (for hardware acceleration via CUDA). Training and validation (using a test data set of 1,000 points) took around 12 hours and was done using mini-batches of 50 points and 50 epochs. Increasing the epoch count to 100 or 200 may have resulted in lower test error, but at the cost of additional computational burden.

2.5. *Random Forest*

Since the monumental study by King et al. (1995), new methods have come to the forefront of machine learning. Among these are the tree based methods *bagging*, *random forest*, and *boosting*, where a large number of decision trees are combined together to improve prediction accuracy.

Here, we consider the random forest learning algorithm, which uses less parameters than boosting and includes bagging as a special case. Random forest operates under two parameters: the number of trees and the number of predictors m to consider at each split in the tree. Two common ways of choosing m are to set it equal to the square root of the number of predictors, $m = \sqrt{p}$, or the by taking the base-2 logarithm, $m = \log_2 p$.

10-fold cross-validation was used to select the number of trees and decide between these two cases of m. The dataset consisted of 180 predictors (100 true), while the number of samples was chosen from the set $n \in \{20, 180, 1200\}$, corresponding to the cases where $n \ll p$, $n \approx p$, and $n \gg p$. The number of trees was varied from 1 to 5000, and the corresponding misclassification rates can be found in Figure 2.

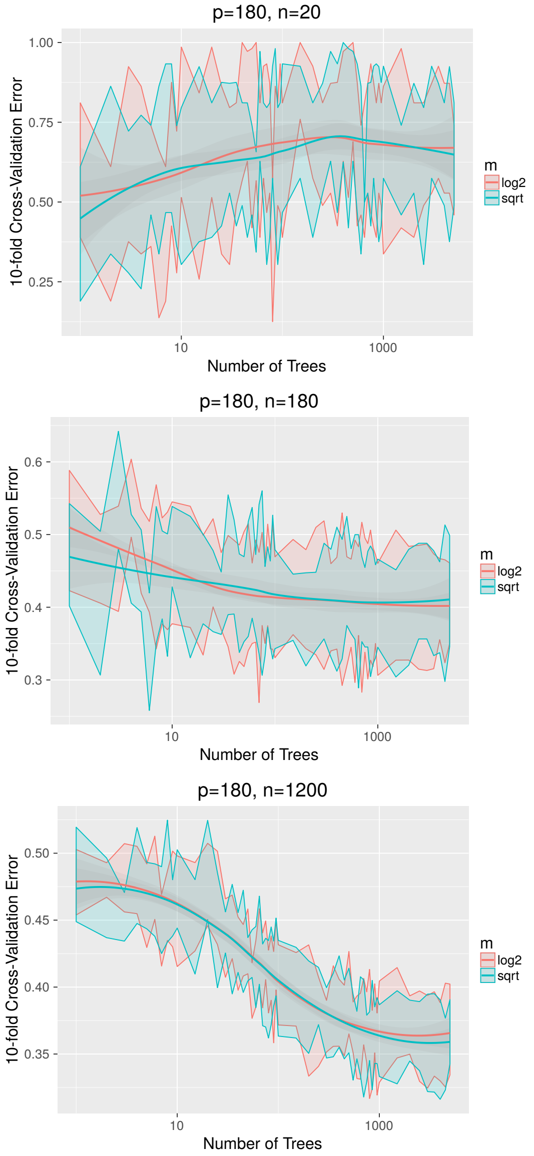

Figure 2. 10-fold cross validation error for three cases of n, the number of trees, and p, the number of predictors. The shaded regions represents the 95% ($2\hat{\sigma}$) confidence interval on the mean.

Overall, we found similar results for the two cases of m; therefore we follow the standard practice and set $m = \sqrt{p}$. When it comes to the number of trees, the misclassification rate tends to flatten out after $N_{\text{tree}} = 1000$ for the cases $n \in \{180, 1200\}$. For the case $n \ll p$, however, having more trees appears to result in misclassifications rates worse than random guessing, perhaps due to overfitting the training dataset. Since 1000 trees was roughly the optimal amount in 2 of the 3 cases, we select this as the number of trees in our model.

3. APPLICATION

After the conclusion of the simulation study, the validity of the results will be confirmed using ARCENE mass-spectrometric data (Guyon et al. 2004) and Parkinson's disease data (Little et al. 2007) from the UCI machine learning repository. The former set of data has continuous features, high-dimensionality, and a relatively small number of instances, while the latter has low-dimensionality and a relatively large sample size. The ARCENE data was originally prepared for the 2003 NIPS feature selection challenge, and is a popular high-dimensional teaching tool within the academic community. The Parkinson's disease data was compiled by University of Oxford researchers in collaboration with the National Center for Voice and Speech, and its original purpose was to study feature extraction for voice disorders. Findings that hold true under these real-world data sets as well as the simulated data will provide insight regarding the appropriate initial classification method for a given binary setting.

4. RESULTS

4.1. *Simulation Study*

The results of the simulation study are summarized in Figures 3 through 8. Each figure shows, for a given value of n, the model test errors for six true values of p. The six sub-plots in each figure illustrate the trajectories of the test errors as additional predictors are added to the models. Ten replications of the simulations were carried out to obtain estimates of the standard errors, as displayed in the figures. Some general trends can be observed from the series of figures. In particular, the random forest method performs quite poorly in all settings. Its tendency to produce test errors oscillating around 0.5 when the true p is not equal to 10 is problematic since this is analogous to guessing. Logistic regression and regularized logistic regression had similar performance when $n > p$, but as p becomes greater than n, regularized logistic regression established itself as the superior method. This behavior was expected due to the restrictions of logistic regression and propriety of regularized logistic regression when $p > n$. By shrinking the coefficient estimates, regularized logistic regression decreases the variance and increases the bias. This increase in bias is out-weighted by the decrease in variance, thus resulting in an overall decrease in the test error. Larger sample sizes induced a decrease in the variability of the test error estimates for the three methods mentioned above. However, the stability of the estimates for neural networks and support vector machines increases as the number of predictors in the generative models increase relative to the bounds of the specified subsets.

Figures 3 and 4 suggest that the estimates of the test error become more stable as the number of parameters increases in a small sample setting. As the stability increases, the test errors tend to diverge further from the Bayes error rate, indicating a negative trend in predictive performance. This decrease in variability uncloaks similar test errors between all five methods. Regularized logistic regression performs quite well when the true p is less than the sample size, but as true p surpasses n, neural networks provide the smallest test errors.

The simulation results from large sample settings are delineated in Figures 5 through 8. When the true p is relatively small, both support vector machines and neural networks exhibit unexpected behavior in the plots. Both methods have test errors below the Bayes error rate, and extremely inaccurate estimates of these test errors. Neural networks seem to perform better when underfitting since the test errors increase as more predictors are added, regardless of the true p. Alternatively, regularized logistic regression has high test error when underfitting, but maintains a constant, lower test error when p is greater than or equal to the true p. Due to the subset of predictors being tested, the issue of underfitting or overfitting does not have pronounced impact on SVM and random forest when the true p is large.

4.2. *Real World Datasets - Parkinsons and ARCENE*

The ARCENE training and validation datasets were combined, resulting in 200 observations with 10000 predictors. Examining our simulation results in this high dimensional space, we expect neural networks and regularized logistic regression to perform best, while logistic regression is likely to perform rather poorly.

For each method, we performed 10-fold cross-validation on the data, computing the mean misclassification rate and its corresponding uncertainty. Our results are summarized in Table 1.

Neural network and regularized logistic regression performed as expected, producing relatively low misclassification rates, while logistic regression performed poorly, consistent with random guessing; SVM performed best. Perhaps most surpising was the performance of random forest: having performed no better than random guessing over most of our simulation, it suddenly rivals the other models in prediction accuracy.

In contrast to ARCENE, the Parkinsons dataset was a relatively low dimensional space, having 195 observations with only 22 predictors. Looking at the simulation study, the neural network and SVM models outperformed the Bayes error in this regime, suggesting something else going on with these models. Assuming these small errors are indicative of true performance, however, we would roughly expect the logistic, regularized logistic, neural network, and SVM models to perform about the same on the Parkinsons dataset, with random forest lagging behind.

Our 10-fold cross validation results are summarized in Table 2. We find that logistic, regularized logistic, neural network, and SVM all perform about equally well, as we would expect. Somewhat surprisingly, however, random forest is the clear winner on the Parkinson's dataset by a wide margin, with the other models all obtaining approximately the same test error.

Table 1. ARCENE

Method	Mean Error	Uncertainty
Logistic Regression	0.470	0.138
Neural Network	0.157	0.056
Random Forest	0.149	0.105
Regularized Logistic	0.105	0.064
Support Vector Machine	0.090	0.039

Note—10-fold cross-validation results on the combined ARCENE test and validation datasets, with 200 observations and 10000 predictors. Misclassification rates were averaged over each of the folds, and the uncertainty found by taking the standard deviation of the 10 misclassification rates.

Table 2. Parkinsons

Method	Mean Error	Uncertainty
Logistic Regression	0.170	0.110
Neural Network	0.255	0.072
Random Forest	0.093	0.022
Regularized Logistic	0.252	0.059
Support Vector Machine	0.201	0.097

Note—10-fold cross-validation results on the Parkinsons dataset, with 195 observations and 22 predictors. Misclassification rates were averaged over each of the folds, and the uncertainty found by taking the standard deviation of the 10 misclassification rates.

5. CONCLUSION

While the results for logistic and regularized logistic regression followed expectation, the remaining models achieved somewhat puzzling results in a number of cases. Part of this is likely due to overfitting or poor tuning, but in some cases there appears to be a more subtle, underlying issue. Results from the simulation study carried over for the ARCENE dataset, but diverged by quite a bit in the Parkinsons dataset.

We had originally planned to also simulate under various alternative settings with multicollinearity present, categorical predictors, and non-linear true relationships, but the scope of the study became unrealistically large. Therefore, introducing these characteristics would be an interesting future research project.

REFERENCES

Guyon, I., Gunn, S. R., Ben-Hur, A., & Dror, G. 2004, NIPS

James, G., Witten, D., Hastie, T., & Tibshirani, R. 2014, An Introduction to Statistical Learning: With Applications in R (Springer Publishing Company, Incorporated)

King, R. D., Feng, C., & Sutherland, A. 1995

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, Advances in Neural Information Processing Systems 25, 1097

Little, M., McSharry, P., Roberts, S., Costello, D., & Moroz, I. 2007, BioMedical Engineering OnLine

Nesterov, Y. 1983

Silver, D., Huang, A., Maddison, C. J., et al. 2016, Nature, 484

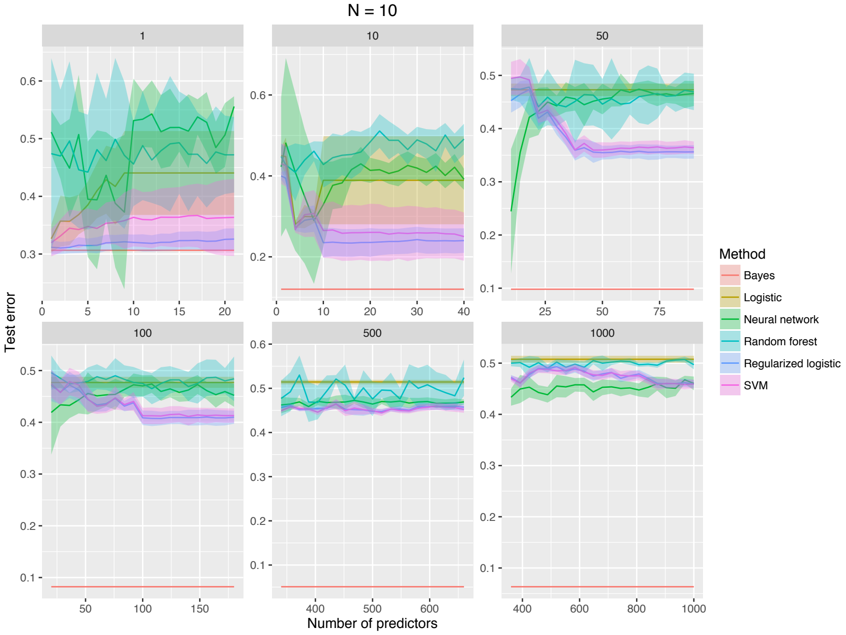

Figure 3. The six plots above compare the logistic regression, regularized logistic regression, random forests, and SVM to the Bayes error rate for the six combinations of p and $n = 10$.

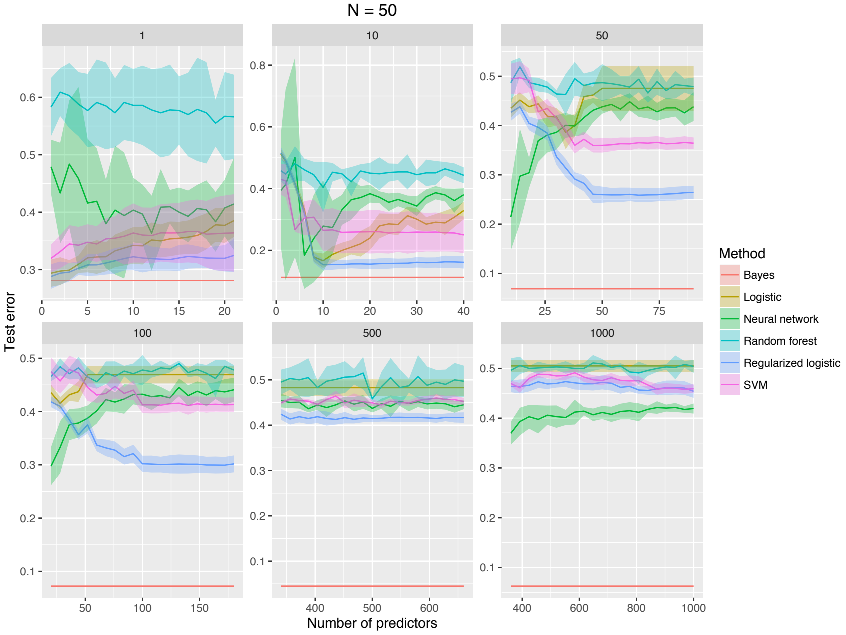

Figure 4. The six plots above compare the logistic regression, regularized logistic regression, random forests, and SVM to the Bayes error rate for the six combinations of p and $n = 50$.

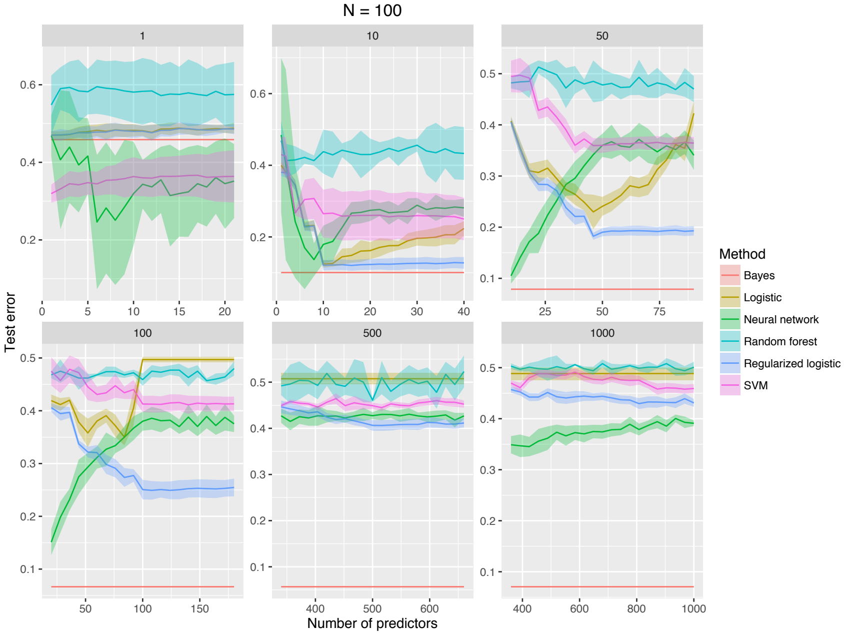

Figure 5. The six plots above compare the logistic regression, regularized logistic regression, random forests, and SVM to the Bayes error rate for the six combinations of p and $n = 100$.

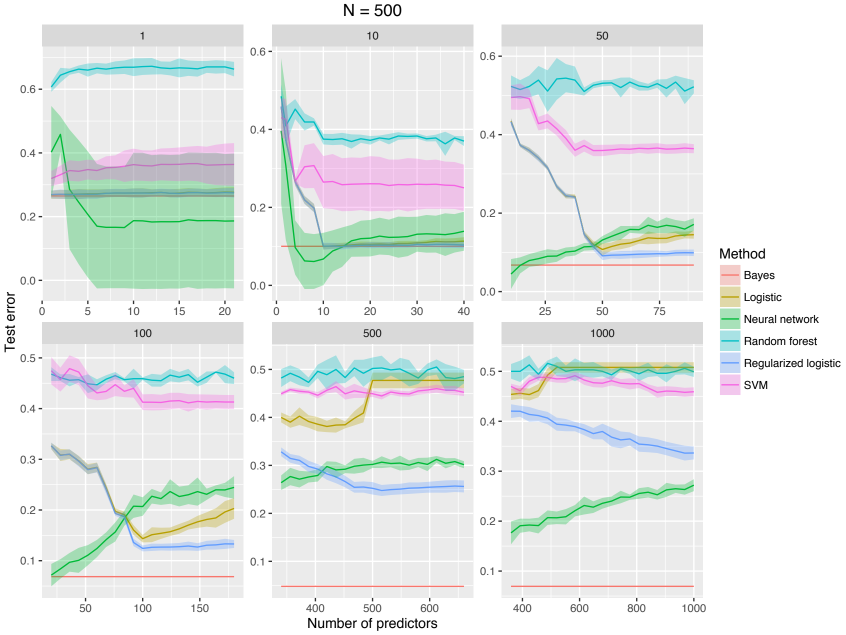

Figure 6. The six plots above compare the logistic regression, regularized logistic regression, random forests, and SVM to the Bayes error rate for the six combinations of p and $n = 500$.

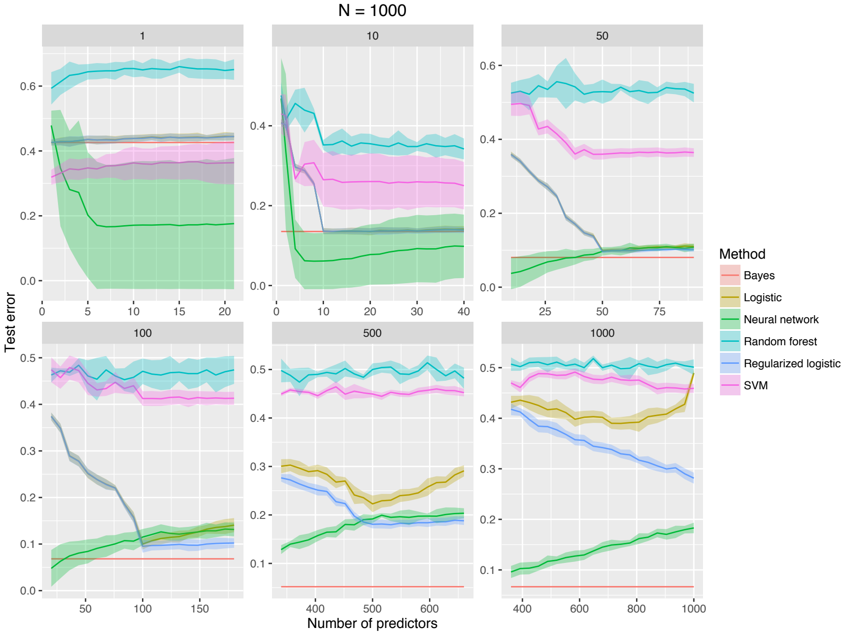

Figure 7. The six plots above compare the logistic regression, regularized logistic regression, random forests, and SVM to the Bayes error rate for the six combinations of p and $n = 1000$.

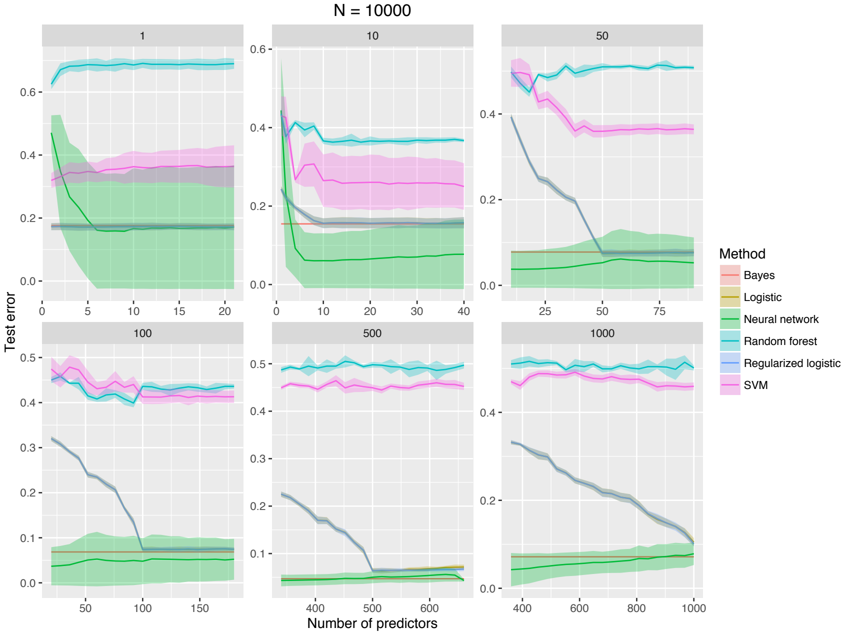

Figure 8. The six plots above compare the logistic regression, regularized logistic regression, random forests, and SVM to the Bayes error rate for the six combinations of p and $n = 10000$.