STATS 202- Final Project

Yifan Wu

8/9/2016

"Talk is cheap. Show me the code."

-- by Linus Torvalds

Table of Contents

INTRODUCTION 2

DATA SELECTION AND PROCESSING 2

MODELLING 3

1. LOGISTIC REGRESSION 3
2. NAÏVE BAYES ALGORITHM 3
3. K-NEAREST NEIGHBORS (KNN) ALGORITHM 3
4. ADAPTIVE BOOSTING (ADABOOST) WITH DECISION TREE 4
5. GRADIENT BOOSTING MACHINE (GBM) 5
6. RANDOM FORESTS ALGORITHM 6
7. SUPPORT VECTOR MACHINE WITH LINEAR KERNEL 7
8. SUPPORT VECTOR MACHINE WITH RADIAL KERNEL 8
9. NEURAL NETWORKS 9

EVALUATION 11

DISCUSSION 12

Introduction

The objective of this project is to predict whether the URL is relevant for the query users enter or not. A comparison of 9 different machine learning algorithms has been investigated to achieve the best prediction performance. It turns out that the best 3 models that achieved the highest accuracy based on a 5-fold cross validation are Neural Networks, Gradient Boosting Machine and Support Vector Machine with radial kernel.

Data Selection and Processing

The response variable is relevance, and a training data set with 10 predictors and 80,046 observations is provided. Since the objective is for prediction, I decided to include all 10 attributes into the model. Due to computational limits, I separated the full data into two sets with 36384 observations in set A and the remaining 43662 in set B. All 5-fold cross validation (CV) is performed using data from set A.

From the integrated correlation and histogram plot (on page 3), it is clear to observe that sig 3 and sig 5 have a high correlation (0.81) with each other. It is also worth to note that sig 3,4,5,6 have a highly skewed distribution from the histogram. It might be helpful to center or regularize the data before analyzing them. Hence, I decided to scale these variables to mean 0 with standard deviation 1.

I also found that there were only 3930 observations (4.9% of total) with query length greater than 6, so I transformed query length greater than 6 into a new category “6+” and converted the query length variable into a factor with 6 levels, 1, 2, 3, 4, 5 and 6+. The table below shows the distribution of query length which I believe it is more appropriate than the original data. It is also clear to see that query length of 1 has 40.2% relevance while query length of 2 has 46.3% relevance. It might suggest that query length is an important variable for prediction.

Query length	Not relevant	Relevant	Total	% of Relevant
1	11556	7761	19317	40.2%
2	14406	12443	26849	46.3%
3	9525	7640	17165	44.5%
4	4969	3836	8805	43.6%
5	2283	1697	3980	42.6%
6+	2320	1610	3930	41.0%

Modelling

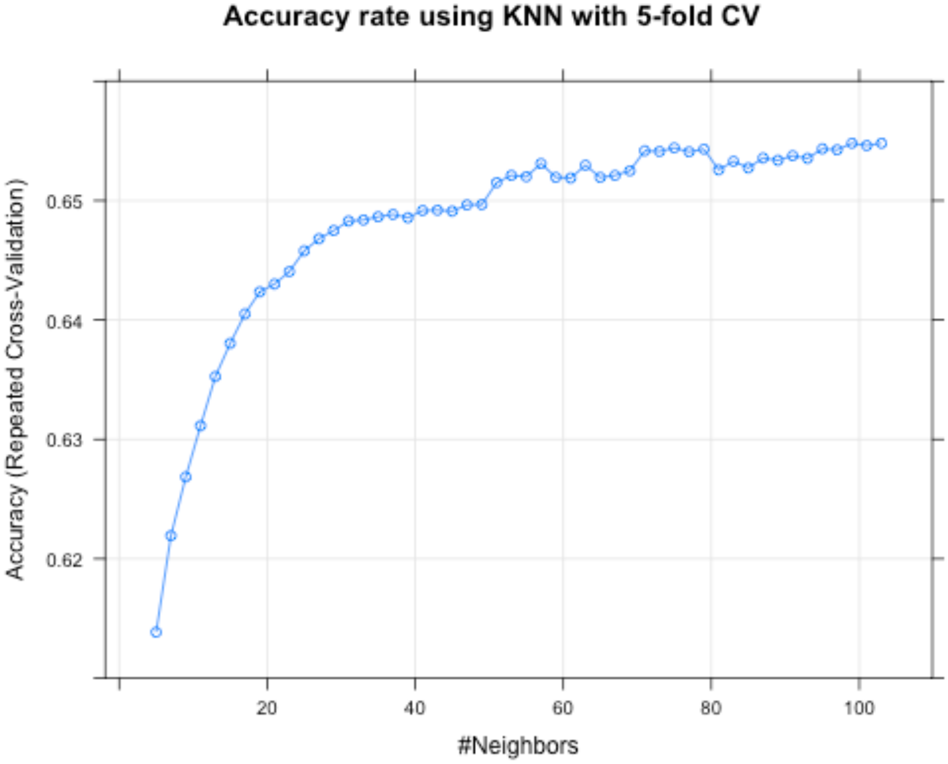
1. Logistic Regression

The very first algorithm I try is logistics regression, which measures the probability that Y belongs to a certain category. There are no tuning parameters required for logistics regression. The accuracy rate reported by 5-fold cross-validation using logistic regression is 65.62%

2. Naïve Bayes Algorithm

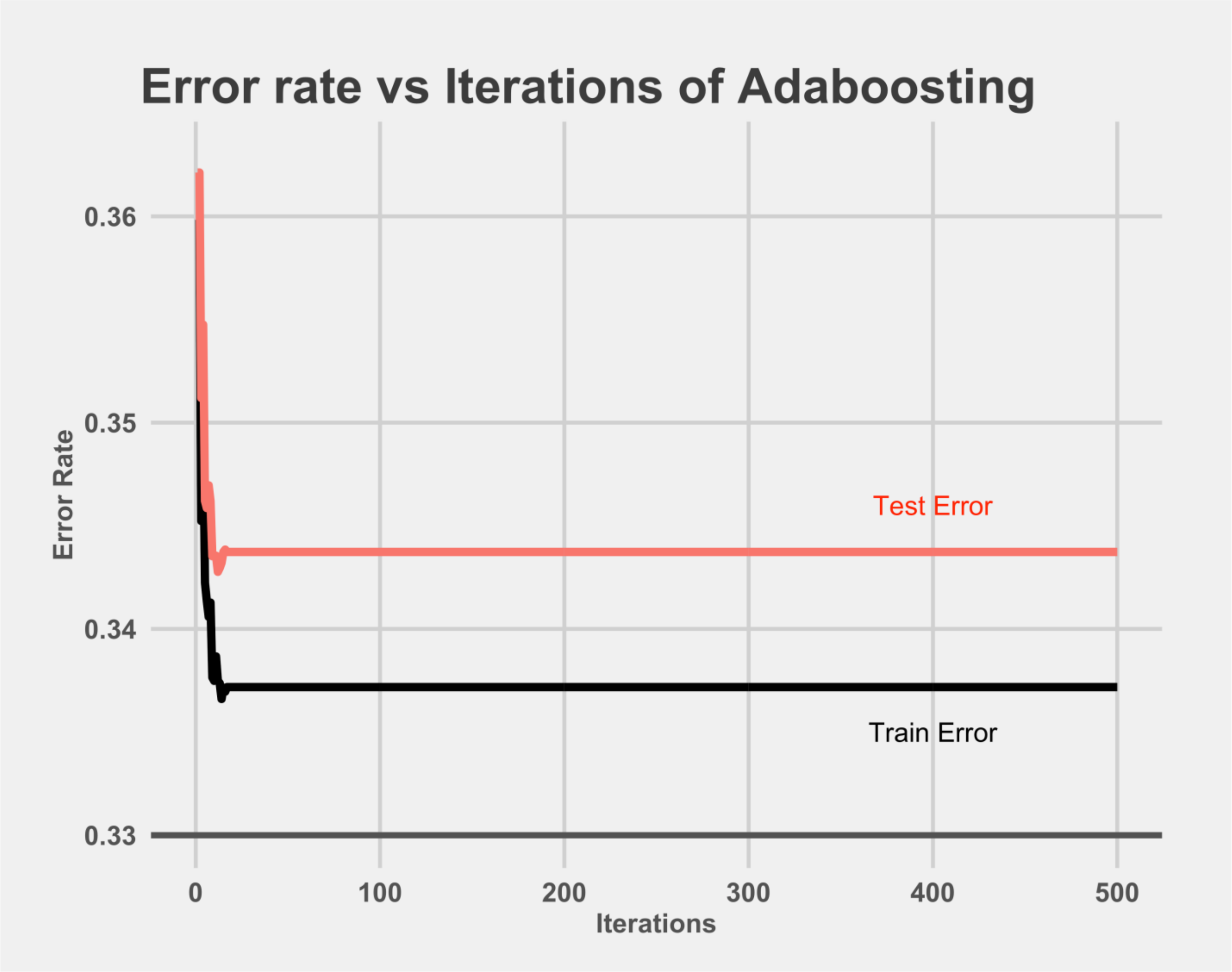
Naïve Bayes algorithm computes the conditional-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule. The naive Bayes makes the assumption that the predictors are independent. This may not hold in our data set which in turn might lower the final prediction accuracy. I have tried two kernel options which are estimated kernel density and using normal density for the kernel. The accuracy rate with 5-fold CV for the normal density is 60.76% and accuracy rate for the estimated kernel density is 58.42%.

3. K-Nearest Neighbors (KNN) Algorithm



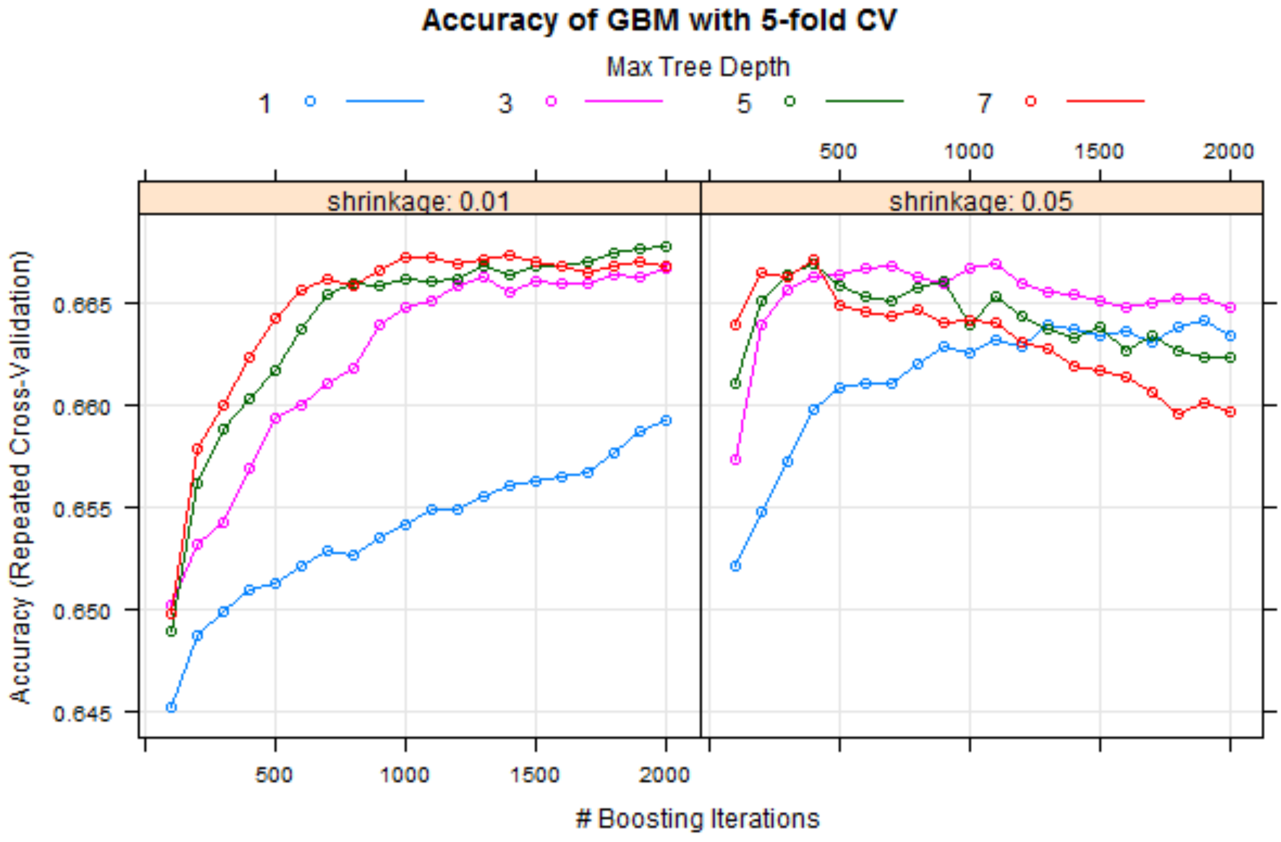
KNN is a commonly used non-parametric algorithm in practice and it only involves one tuning parameters, K. It's critical to choose the optimal K value since a small value of K will give us a highly flexible decision boundary while a larger K will produce a less flexible boundary that is close to linear. According to the KNN figure above, the value of K from 1 to 103 with a step of 2 is plotted. The optimal K chosen here is k=75 with a 5-fold CV accuracy rate of 65.44%.

4. Adaptive Boosting (AdaBoost) with Decision Tree



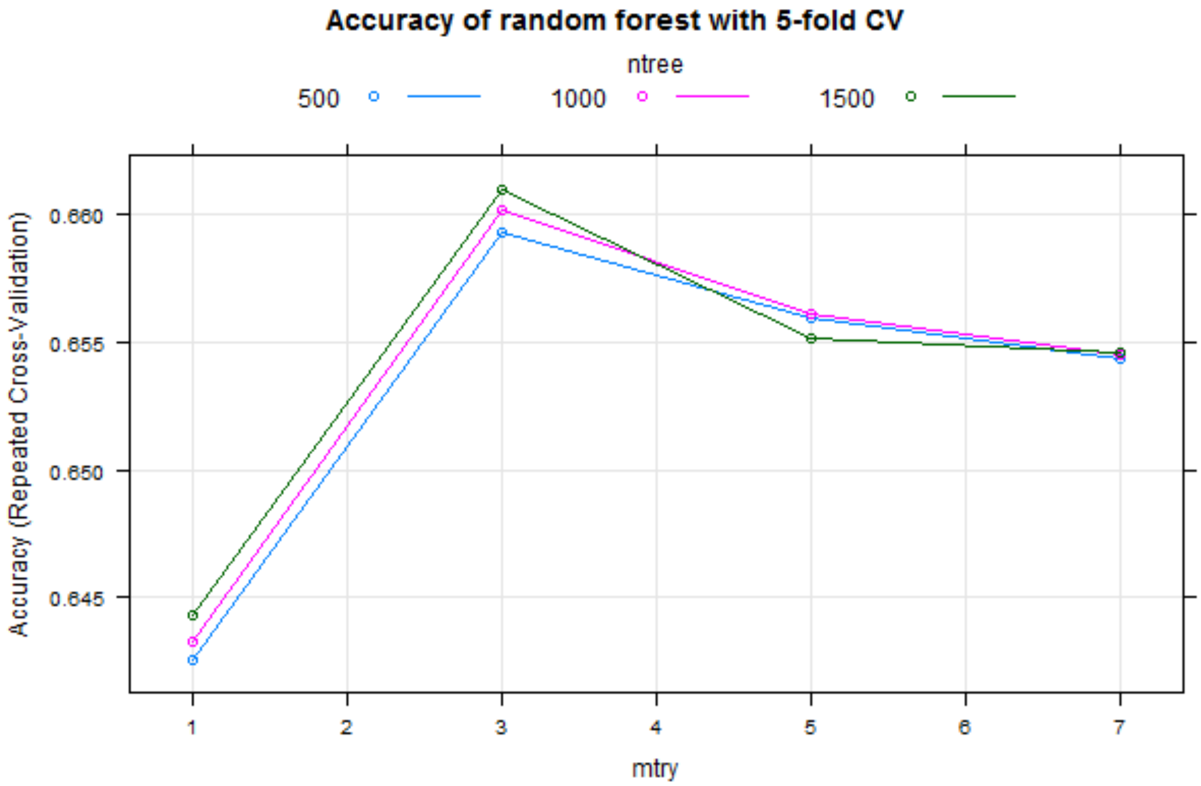
AdaBoost is an algorithm which can be used with many other types of learning algorithms to improve their performance. Decision tree is used here as the weak learner. A plot of training errors and testing errors against no. of iterations is shown above. The training error approaches to 0.337 pretty quickly after the first few iterations and test error also reaches the minimum MSE which is 0.342. The 5-fold CV accuracy rate is 65.72%.

5. Gradient Boosting Machine (GBM)



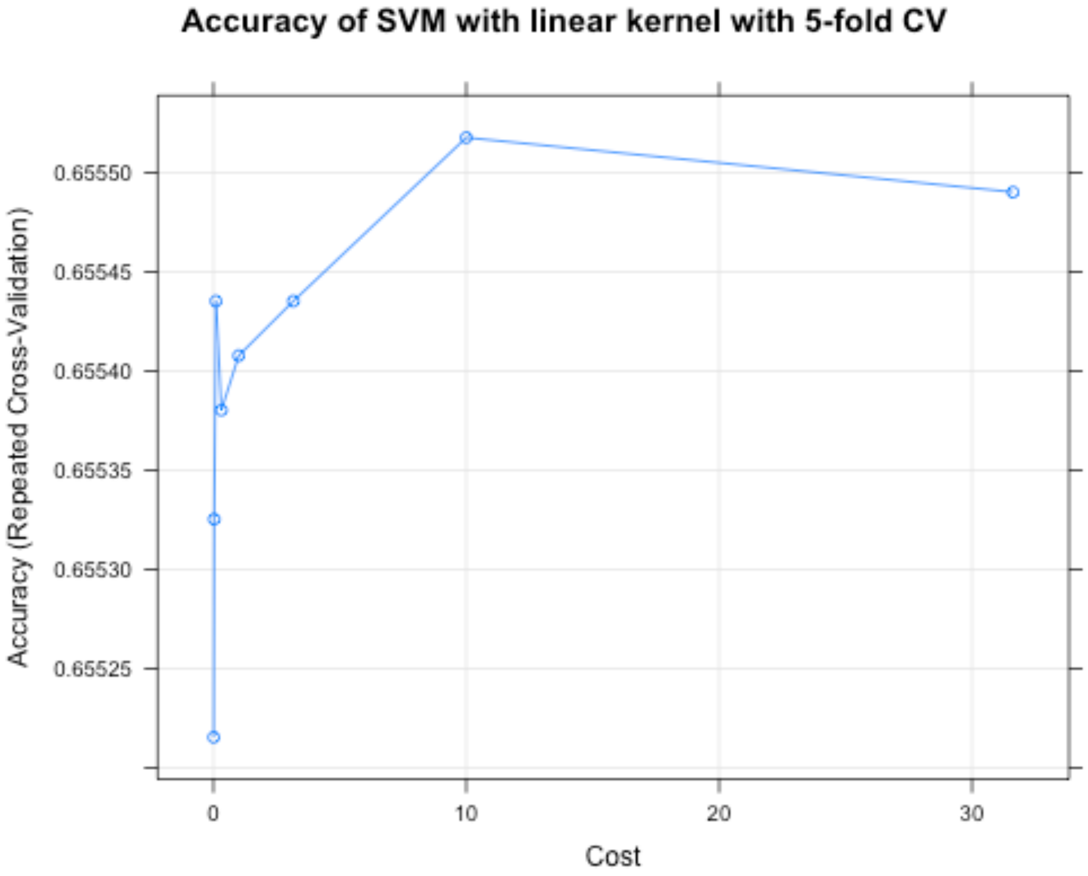
GBM is essentially the same as AdaBoost, which sequentially fits new models to provide a more accurate estimate of the response variable. I used the approach introduced in the textbook and tuned a few parameters. By comparing the shrinkage of 0.01 and 0.05 or left vs right plot, we can tell that a smaller shrinkage performs better. The reason behind is a large shrinkage means a more aggressive learning approach, which might lead to overfitting. The interaction depth refers to the number of splits in each tree. Given the shrinkage of 0.01, depth of 5 performs slightly better than that of 1,3 and 7; a larger number of trees also outperform the smaller one. Hence, the final parameter chosen is shrinkage = 0.01, interaction depth = 5 and trees = 2000. This model gives us a 66.23% accuracy rate in the 5-fold CV.

6. Random Forests Algorithm



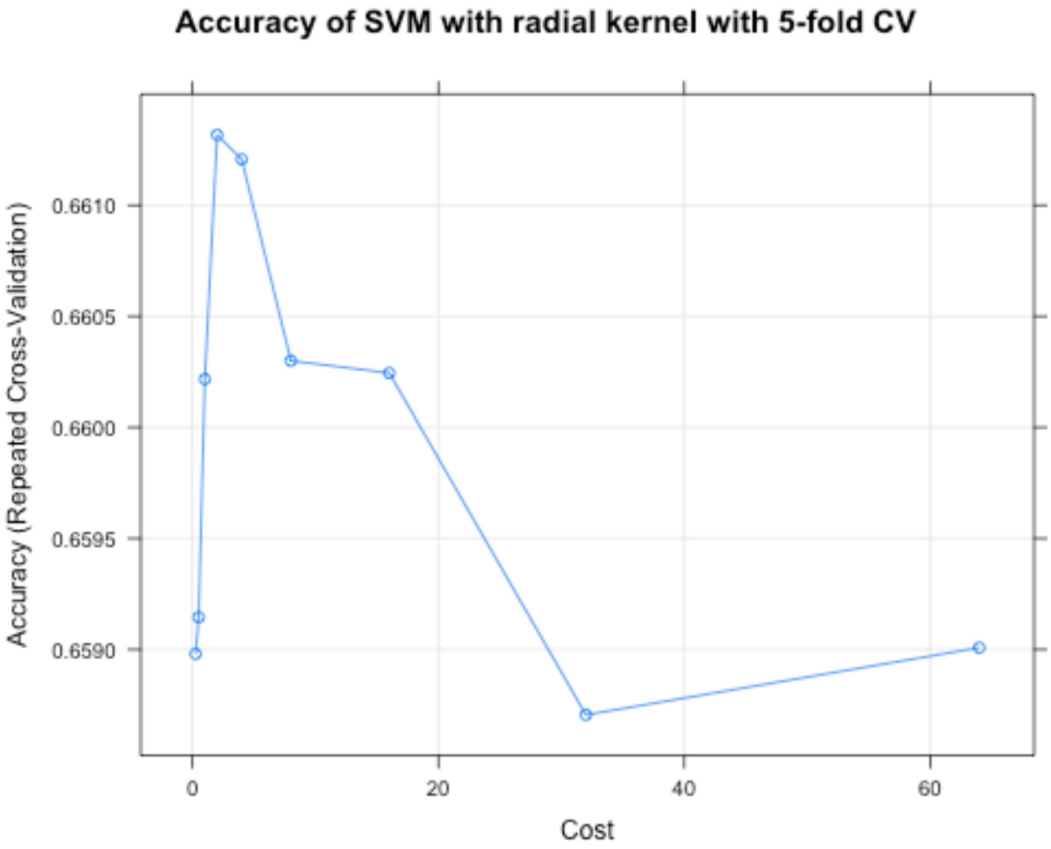
Random forests algorithm is one of the ensemble learning methods which build a set of decision trees and average the outcome of each tree. Empirically, mtry is equal to the square root of number of predictors. I have tried the values of mtry for 1,3,5,7. It turns out that 3 (square root of 10) is the optimal mtry parameter which maximizes the accuracy rate. And the model with number of trees = 1500 performs better than that of 500 and 1000. The final model with mtry = 3 and n.trees=1500 produces an accuracy rate of 66.10% with 5-fold CV.

7. Support Vector Machine with Linear Kernel



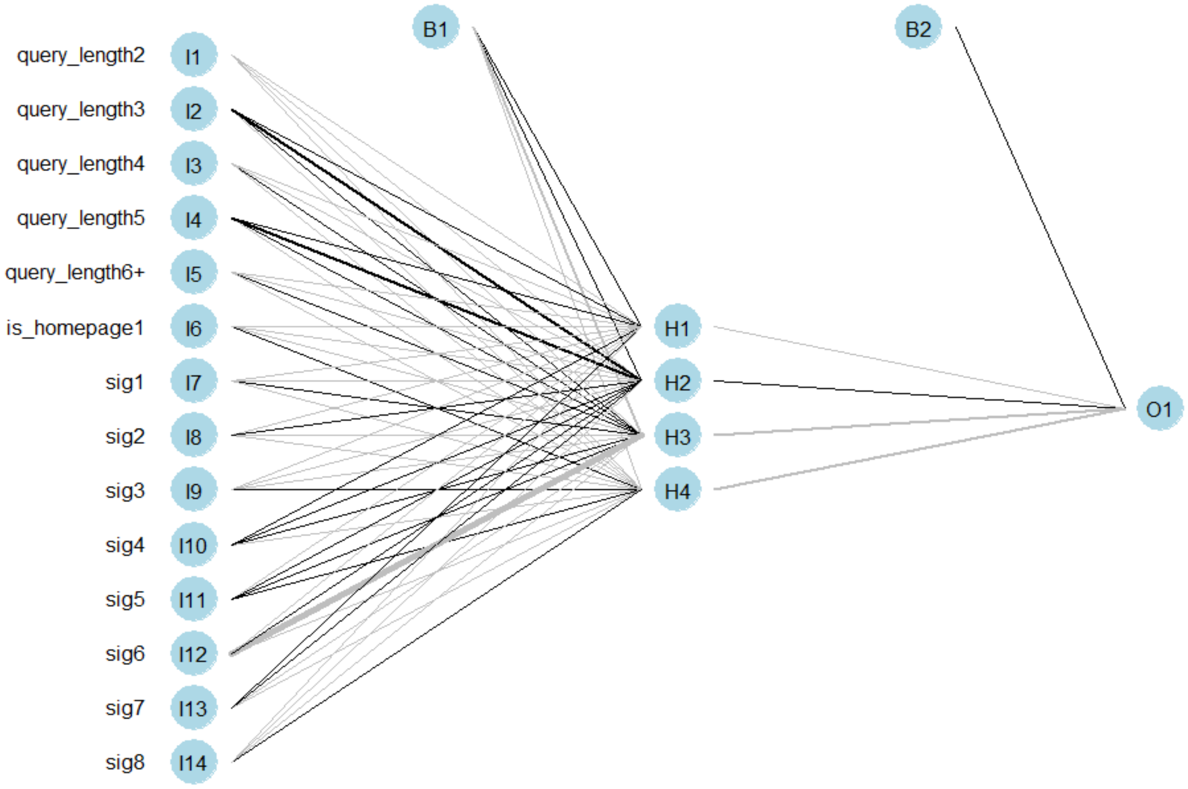
SVM is a supervised learning model which will separate a given set of data by an optimal hyperplane. SVM with a linear kernel assumes a linear boundary to separate the classes. According to the figure above, as the cost increases, the accuracy rate will increase at the beginning, and the curve eventually flattens out after cost =10. Hence, the optimal cost is 10 and the accuracy rate with a 5-fold CV is 65.55%.

8. Support Vector Machine with Radial Kernel

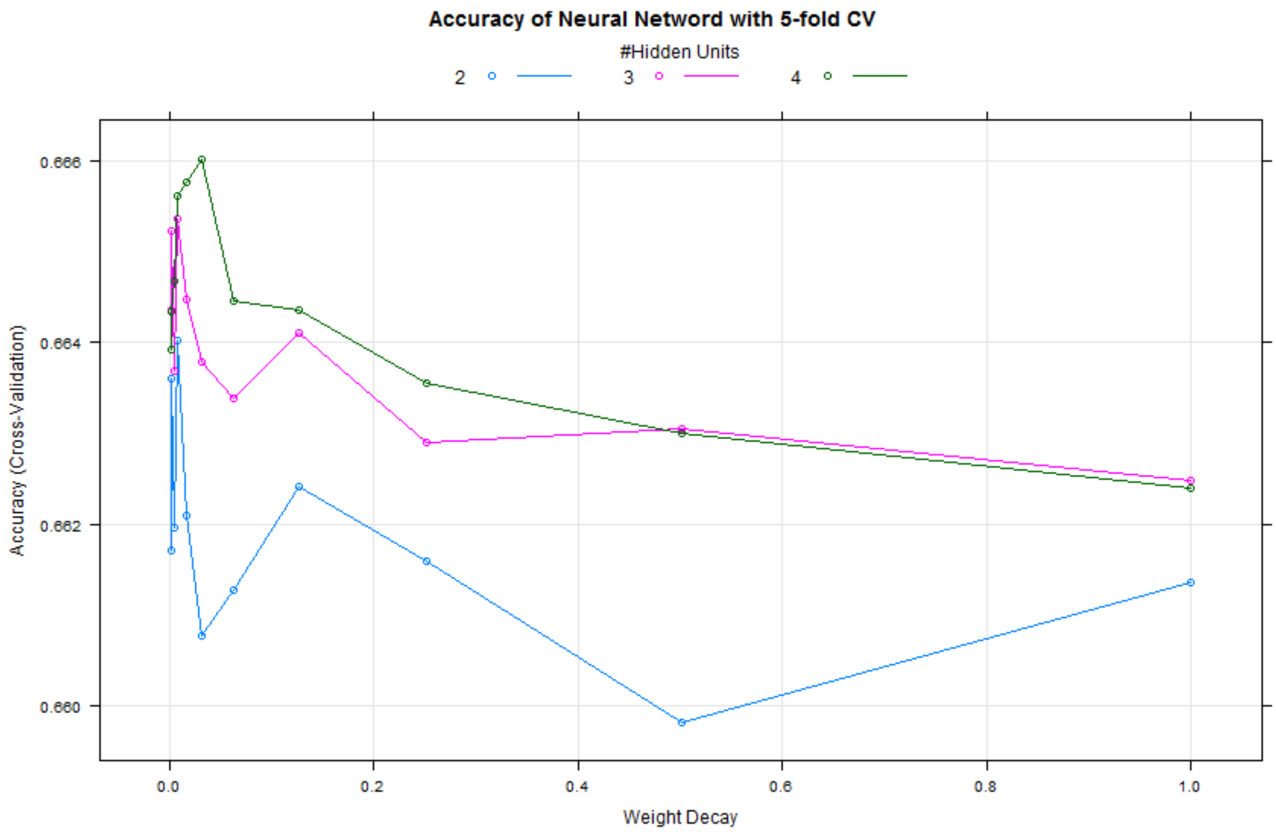


SVM with a radial kernel assumes a non-linear boundary to separate the classes. According to the figure above, as the cost increases, the accuracy rate will increase at the first few points, and the accuracy rate starts to decrease around 18. Holding the sigma constant at 0.0828, the optimal cost is 2 and the accuracy rate with a 5-fold CV is 66.13%.

9. Neural Networks

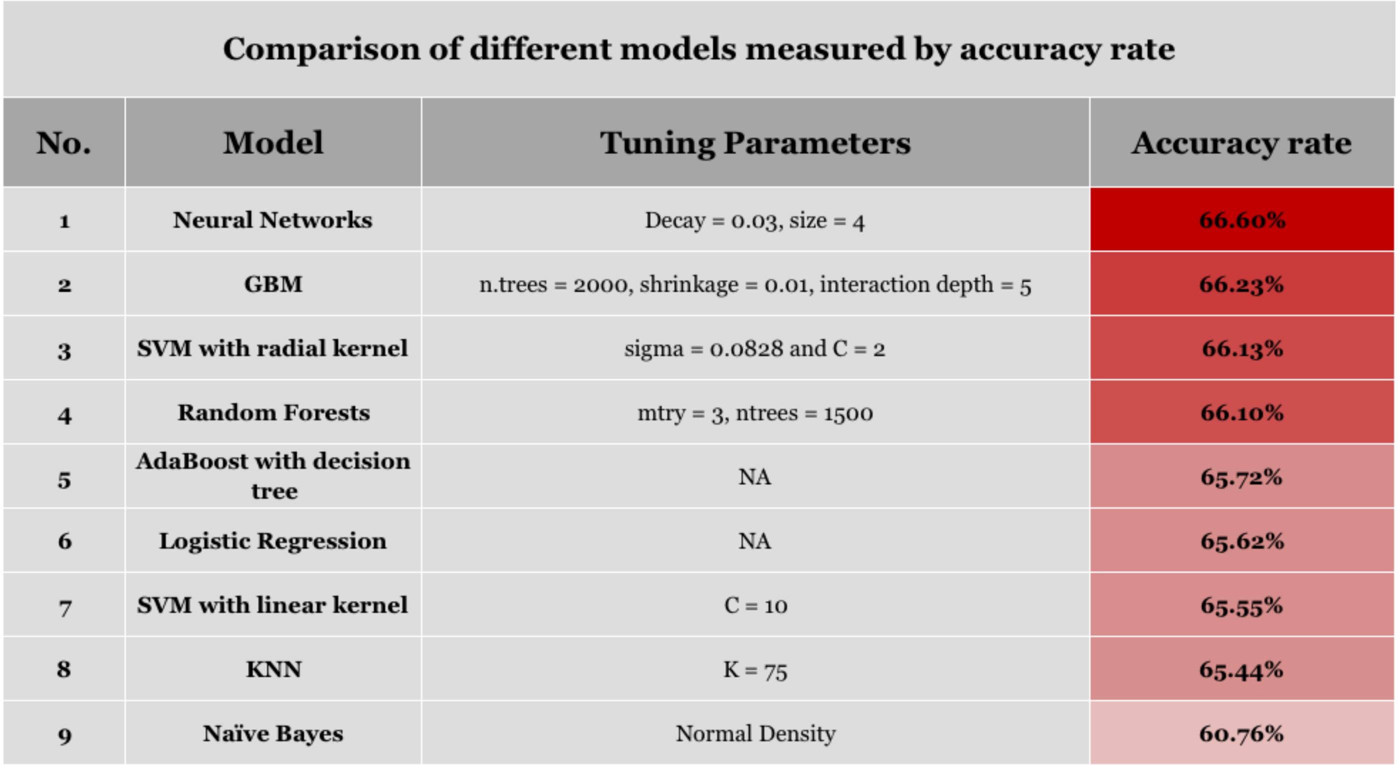


According to the definition on Wikipedia, “An artificial neural network (ANN) learning algorithm, usually called "neural network" (NN), is a learning algorithm that is inspired by the structure and functional aspects of biological neural networks.” Neutral networks are often applied to solve complex and high-dimensional problems such as computer vision, speech recognition and handwriting recognition etc. The reason why I decided to try neural networks is that we can image those thousands of internet pages actually sort of connected to each other which eventually form a giant network. I used the nnet() package in R to implement neural networks. The high level representation of how to fit our data set into neutral networks is visualized in the figure above. The first column refers to the input variables; the second column represents 4 hidden layers defined; and the last column is the output.



Neutral networks have a few tuning parameters, such as hidden layers and weight decay etc. I focus on tuning these two parameters. As we can see in the figure, as the value of decay increases, the accuracy rate goes down. The accuracy rate reaches to the optimal point at a very small decay value. A different hidden layers of 2,3 and 4 are also considered here. It turns out that the neutral networks with 4 hidden layers perform better than the other two. The optimal tuning parameters for neutral networks are hidden layers = 4 and decay = 0.0316. The accuracy rate with 5-fold CV is 66.60%, which is a bit higher than other algorithms.

Evaluation

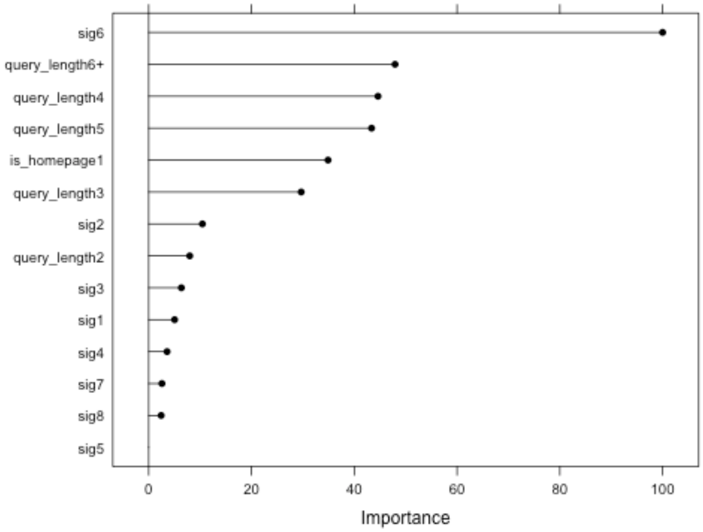


Comparison of different models measured by accuracy rate

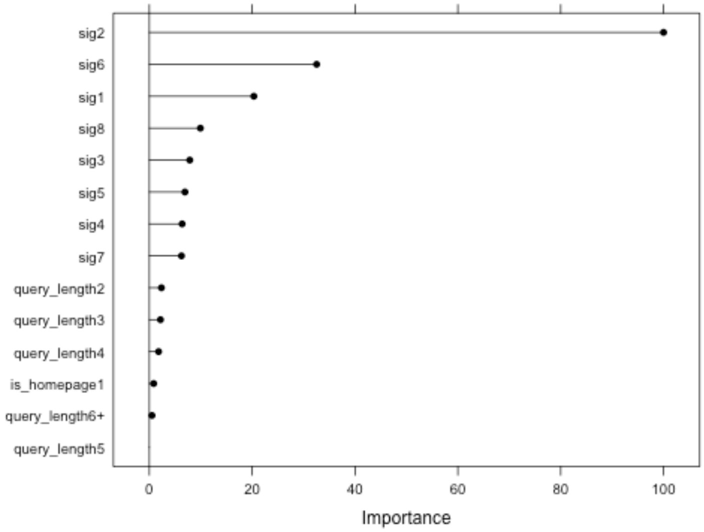
No.	Model	Tuning Parameters	Accuracy rate
1	Neural Networks	Decay = 0.03, size = 4	66.60%
2	GBM	n.trees = 2000, shrinkage = 0.01, interaction depth = 5	66.23%
3	SVM with radial kernel	sigma = 0.0828 and C = 2	66.13%
4	Random Forests	mtry = 3, ntrees = 1500	66.10%
5	AdaBoost with decision tree	NA	65.72%
6	Logistic Regression	NA	65.62%
7	SVM with linear kernel	C = 10	65.55%
8	KNN	K = 75	65.44%
9	Naïve Bayes	Normal Density	60.76%

The table above summarizes 9 different machine learning approaches for this project and it is ranked by 5-fold CV accuracy rate at a descending order. The winner is neutral networks, follow by GBM and SVM with radial kernel algorithm. Let us investigate the variable importance plot of the top 4 algorithms with the highest accuracy rate.

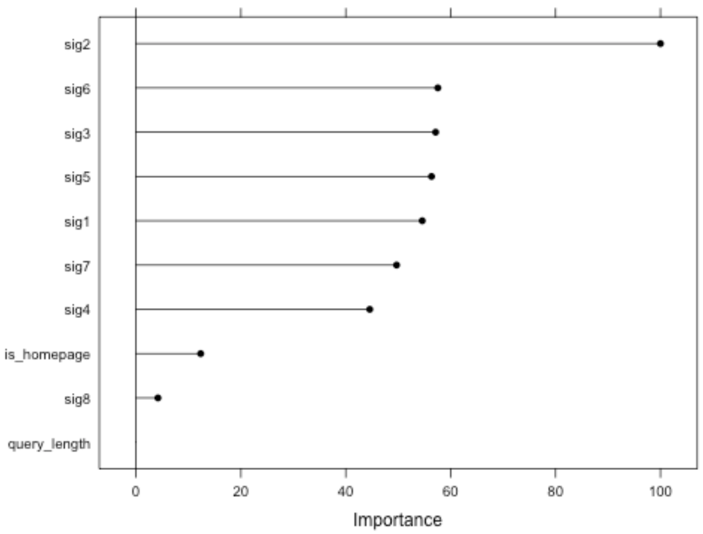
Variable importance plot of Neural Networks



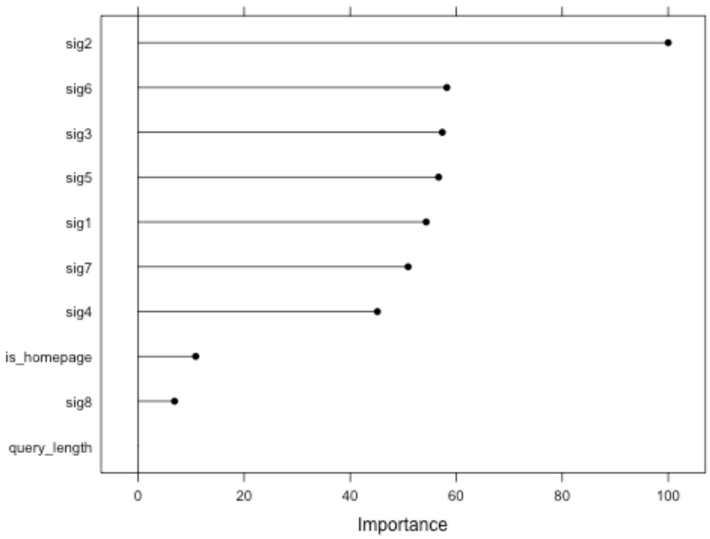
Variable importance plot of GBM



Variable importance plot of SVM with radial kernel



Variable importance plot of Random Forests



It is extremely interesting to observe that different algorithms actually pick up different important variables while achieving a similar accuracy rate. Sig 2 and sig 6 are two common important variables across all 4 algorithms. However, it seems that only neutral networks believe the query length is important while others don't. GBM seems to believe that sig 2 is the single most important variable that has a higher weight than the remaining variables. SVM with radial kernel and random forrests perform almost identical in terms of variable importance.

Discussion

Due to time and computational constraints, I only used a portion of training data and perform a very limited search for tuning parameter. In practice, it took much longer time and efforts to train and tune our models, especially for neutral networks.