PAUL M Girdler
CLASS 15
paulgird@buffalo.edu.au

Homework Assignment 4
EAS506 - Statistical Data Mining 1

2018-11-27

1 Exercise 1

For the prostate data of Chapter 3, carry out a best subset linear regression analysis, as in Table 3.3 (third column from the left). Compute the AIC, BIC, five- and tenfold cross-validation, and bootstrap .632 estimates of prediction error.

The Prostate dataset contains 97 data points with 8 predictors and 1 response variable, **lpsa**, the log of the prostate specific antigen.

1.1 Summary

Describe your findings.

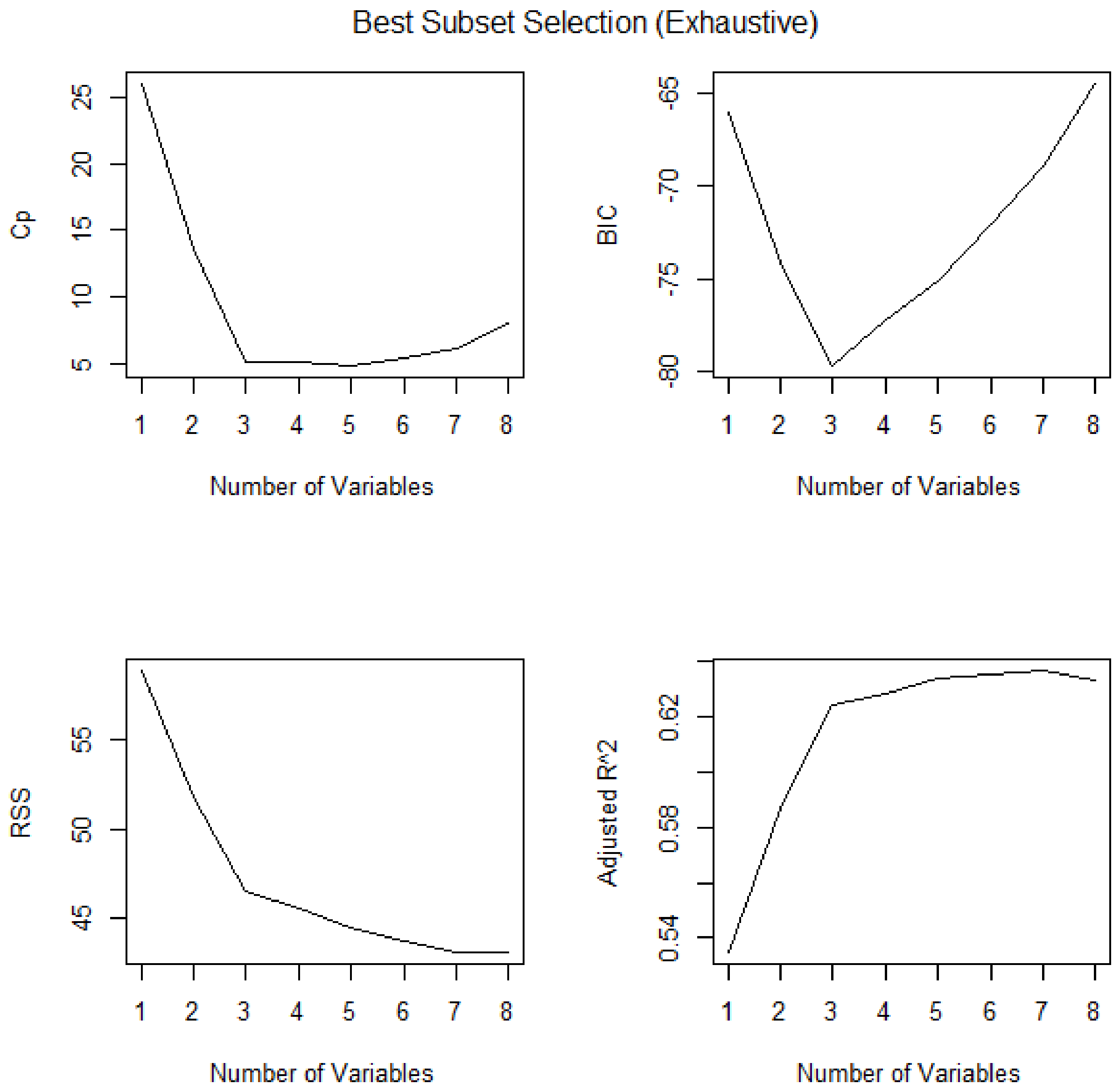
Table: Summary of Results

Method	Number of Variables	Error	Error Type
Best Subset Selection (Exhaustive)	3	0.480	MSE
5-fold Cross Validation	3	0.530	MSE
10-fold Cross Validation	3	0.544	MSE
Bootstrap	7	0.517	0.632 Estimate

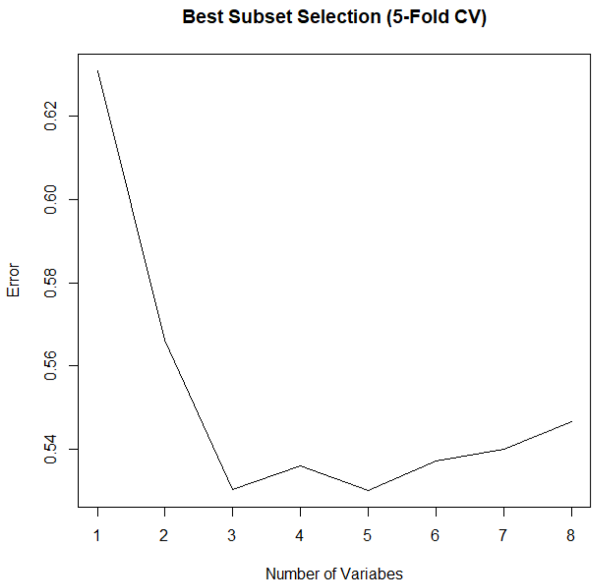
From the summary above it is evident that a model of size 3 was the best in terms of both error, and parsimony. Resampling methods compared favourably to each other. As we would expect, K-fold's method had an upward bias, while 0.632 estimate had a downward bias in error estimate.

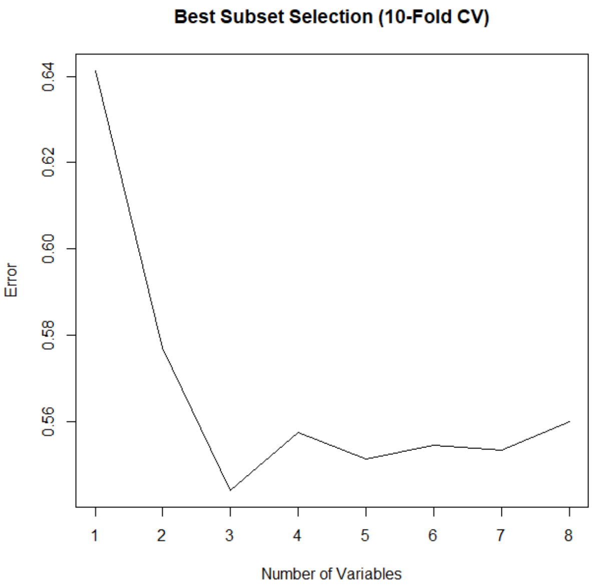
1.2 Method

Exhaustive subset selection was performed. The Cp and BIC were computed for the data. The results are shown below. The Cp and the BIC indicate that the 3 variable model is the best model, in terms of error, and parsimony.

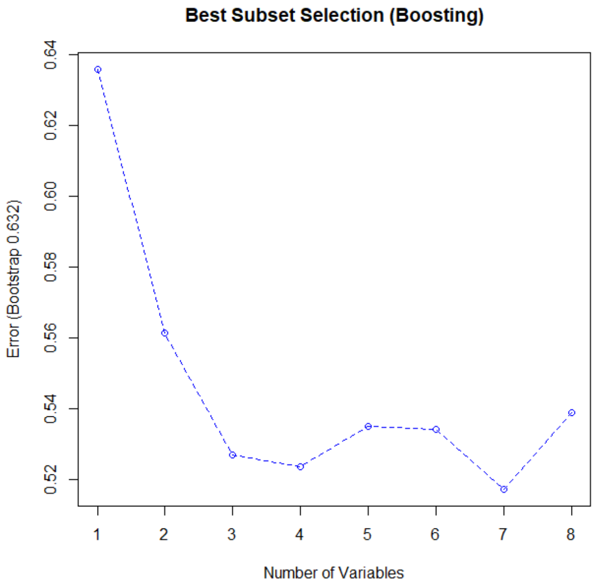


Linear regression for varying model sizes were performed with 10 fold and 5 fold cross validation. The error was computed for each model size. The results are shown below. The error trend indicates the 3 variable model is the best model.





Finally, boosting was performed for subsets of different sizes. The bootstrap prediction error estimation was computed and plotted. The results are shown below. The error trend indicates the 7 variable model is the best (closely followed by the 3 and 4 variable models).



2 Exercise 2

Access the wine data from the UCI machine learning repository (https://archive.ics.uci.edu/ml/datasets/wine). These data are the results of a chemical analysis of 178 wines grown over the decade 1970-1979 in the same region of Italy, but derived from three different cultivars (Barolo, Grignolino, Barbera). The Babera wines were predominately from a period that was much later than that of the Barolo and Grignolino wines. The analysis determined the quantities MalicAcid, Ash, AlcAsh, Mg, Phenols, Proa, Color, Hue, OD, and Proline. There are 50 Barolo wines, 71 Grignolino wines, and 48 Barbera wines. Construct the appropriate-size classification tree for this dataset. How many training and testing samples fall into each node? Describe the resulting tree and your approach.

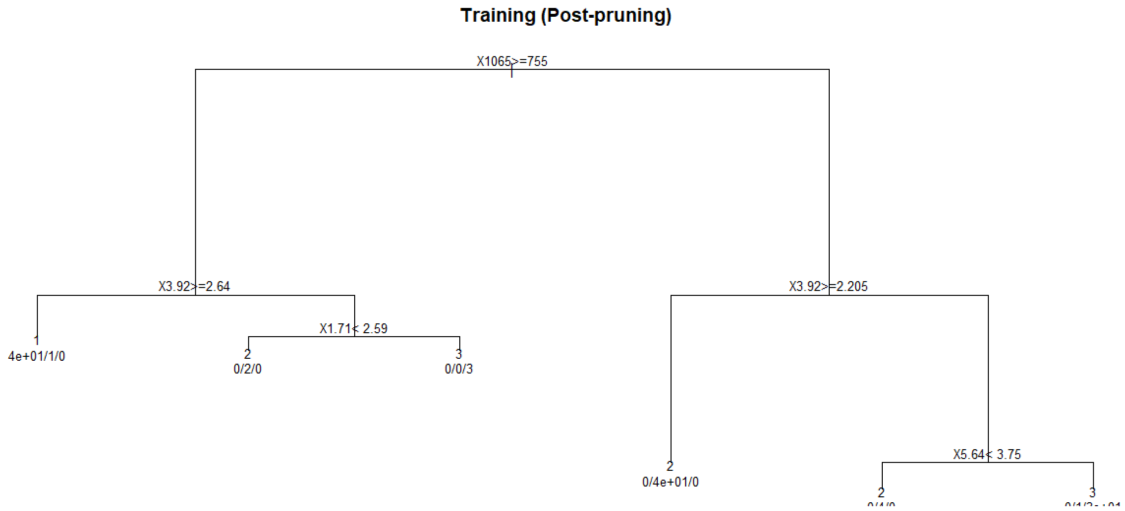
2.1 Summary

Table: Training and Test Samples per Node

Node	Training Samples	Test Samples
Node	Number (Proportion)	Number (Proportion)
1	39 (31.7%)	17 (31.5%)
2	43 (35.0%)	19 (35.2%)
3	2 (1.6%)	3 (5.6%)
4	3 (2.4%)	2 (3.7%)
5	4 (3.3%)	1 (1.9%)
6	32 (26.0%)	12 (22.2%)
Total	123 (100%)	54 (100%)

The table above shows the number of test and training samples falling in each node. It can be observed that the proportion that falls into each node is very similar.

Figure: Pruned-tree



The figure above shows the final pruned tree.

2.2 Methodology

The data was split into test and training. A single tree was grown and then pruned using the minimum Cp value. The test error was calculated at 16.7%. Please note that I did not try to optimize the error rate of the single tree as I believed this not to be the point of this exercise.

To calculate the number of samples in each terminal node the values were aggregated to get a count. The summary function provides this information and more.

3 Exercise 3

Apply bagging, boosting, and random forests to a data set of your choice (not one used in the committee machines labs). Fit the models on a training set, and evaluate them on a test set.

The motive was to compare the performance of different classification methods on the Boston data-set.

3.1 Summary

Describe your findings.

Table: Summary of Results

Method	Parameter	Test Error
RF	$t = 10,000$	4.90%
RF (Bagging)	$t = 10,000$	6.86%
RF (Boosting)	$t = 10,000$, $s = 0.6$	8.78%
Logistic Regression	$p = 12$	11.76%
LDA	$p = 4$	14.71%
kNN	$k = 1$	3.92%

As evident in the table above, KNN outperformed all the other methods in terms of predictive accuracy. All the Random Forest methods outperformed both Logistic Regression, and LDA. The Random Forest method without any bagging or boosting performed better than the other Random Forest methods. It is worth noting that the Random Forest Method had pretty good accuracy, when compared to kNN. However, it should be noted that it is more interpret-able (using various methods such as variable importance) when compared with kNN. Random Forest on on it's own appears very fast to implement and execute.

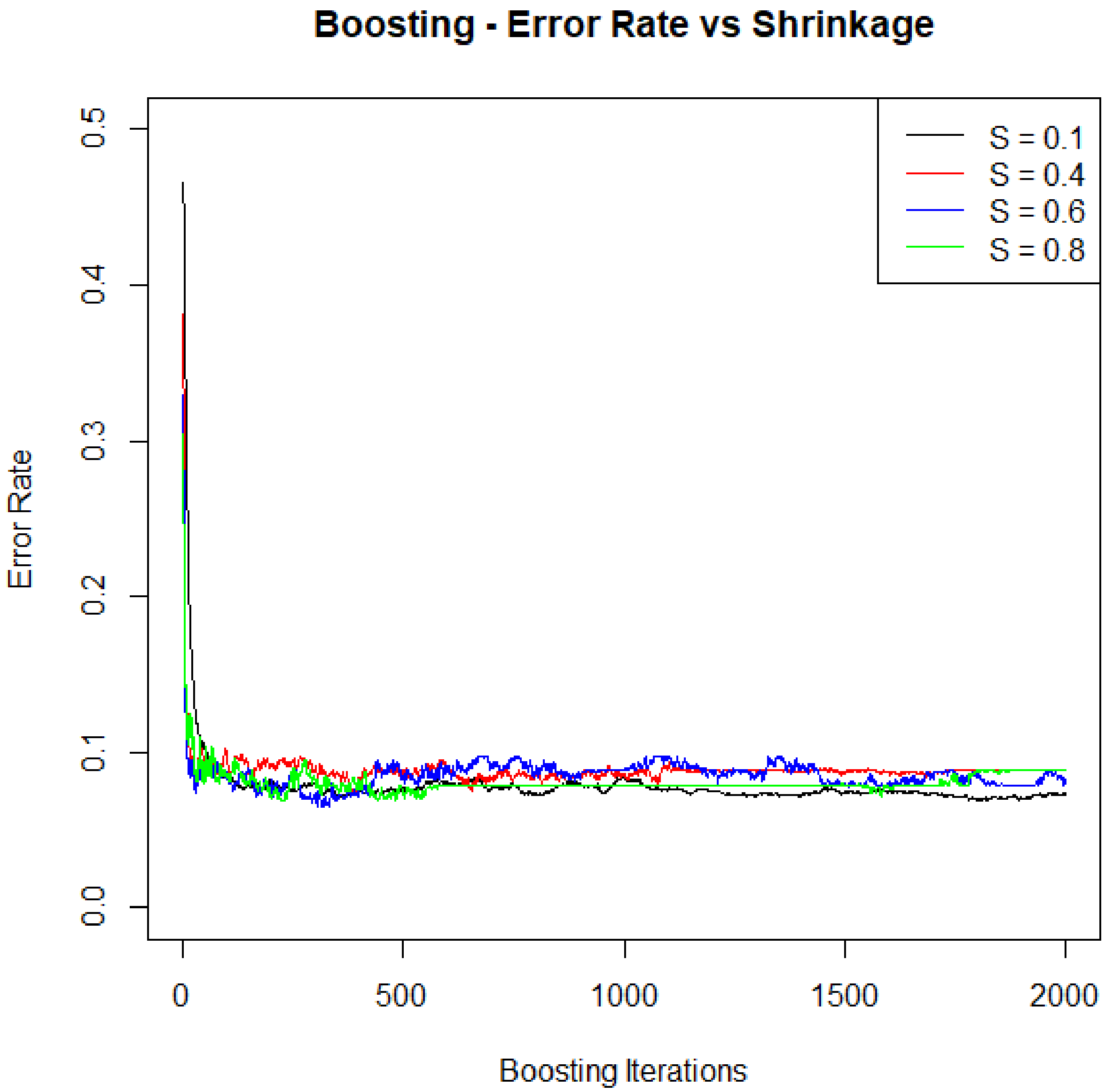
3.2 Methodology

The Boston dataset contains 506 data points with 13 predictors and 1 response variable, **crim**, the per capita crime rate by town. The response variable, **crim**, was converted to a class variable, with two classes:

0 = below median per capita crime rate
1 = above median per capita crime rate

The data, excluding the class, was scaled. The data was randomly split into training and test with a ratio of 4:1 respectively.

Random Forest was performed with 10,000 trees. Random Forest was repeated again but also implementing bagging. Random Forest was again repeated but also implementing boosting. The method was optimized (see figure below). Additionally Logistic Regression, LDA, and KNN were performed. All methods were optimized where appropriate and the associated test errors were computed.


Boosting - Error Rate vs Shrinkage
Error Rate
0.0
0.1
0.2
0.3
0.4
0.5
Boosting Iterations
0
500
1000
1500
2000
S = 0.1
S = 0.4
S = 0.6
S = 0.8