Write up for homework4

Hyungkyu Lim

1) Basically, "prostate" data set is about data to examine the correlation between the level of prostate-specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy. When I used hold out method, the train error is about 0.377 and the test error is 0.846. Based on this method, AIC is about 165.486 and BIC is 191.268.

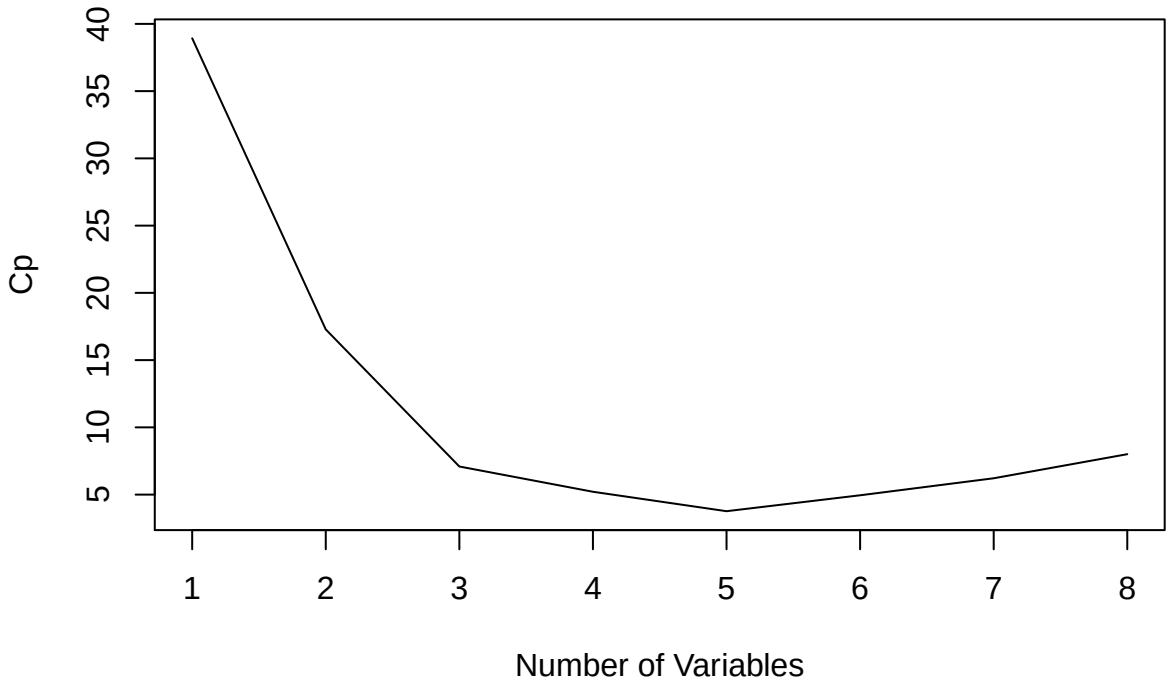
```
## [1] 0.8460985
```

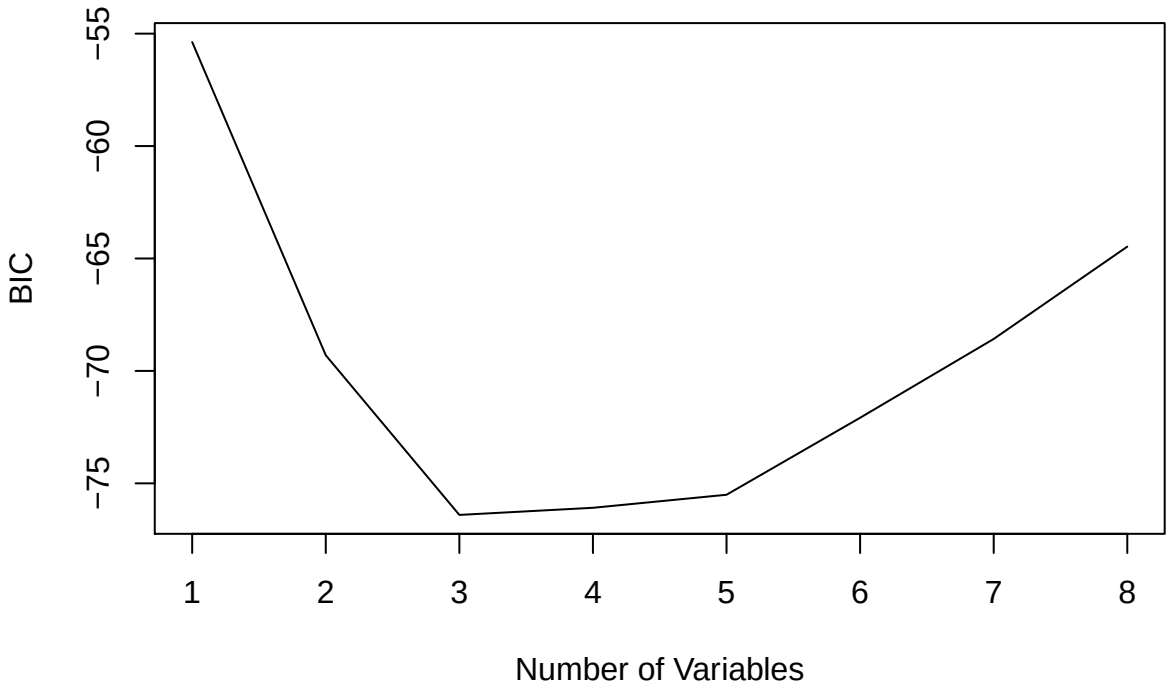
```
## [1] 0.3774099
```

```
## [1] 165.4859
```

```
## [1] 191.2678
```

For exahaustive method, Cp says 5 variables are the best fit and BIC says 3 variable are the best fit. It menas that error is mnimum when there are 5 variables in Cp and there are 3 variables in BIC. Based on the plot for train error and test error, both are the minimum when k=8 of model selection.



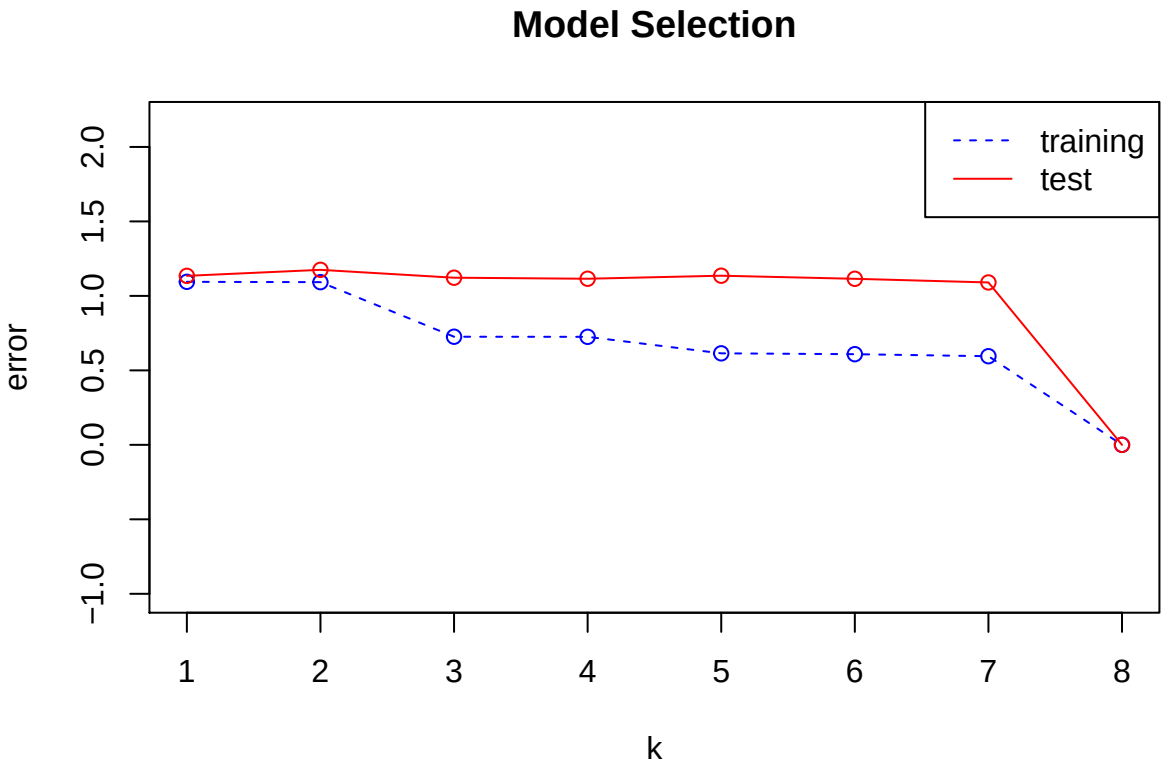



```
## [1] 5
```

```
## [1] 3
```

```
## [1] 38.927840 17.282635  7.088993  5.210842  3.766689  4.949606  6.214480
## [8]  8.005310
```

```
## [1] -55.37293 -69.30207 -76.40353 -76.08786 -75.50710 -72.08369 -68.57746
## [8] -64.47365
```



For cross validation when k=5, when 4th data is test dataset, cross validation error is minimum(about 0.747).

```
## [1] 0.6586470 0.6115945 0.5869321 0.5586724 0.5817036 0.5888256 0.5626422
## [8] 0.5691043
```

```
## [1] 0.8115707 0.7820450 0.7661150 0.7474439 0.7626949 0.7673497 0.7500948
## [8] 0.7543900
```

```
## [1] 4
```

For cross validation when k=10, when 7th data is test dataset, cross validation error is minimum(about 0.747). Although k is changed, the minimum closs validation error is same for k=5 and 10.

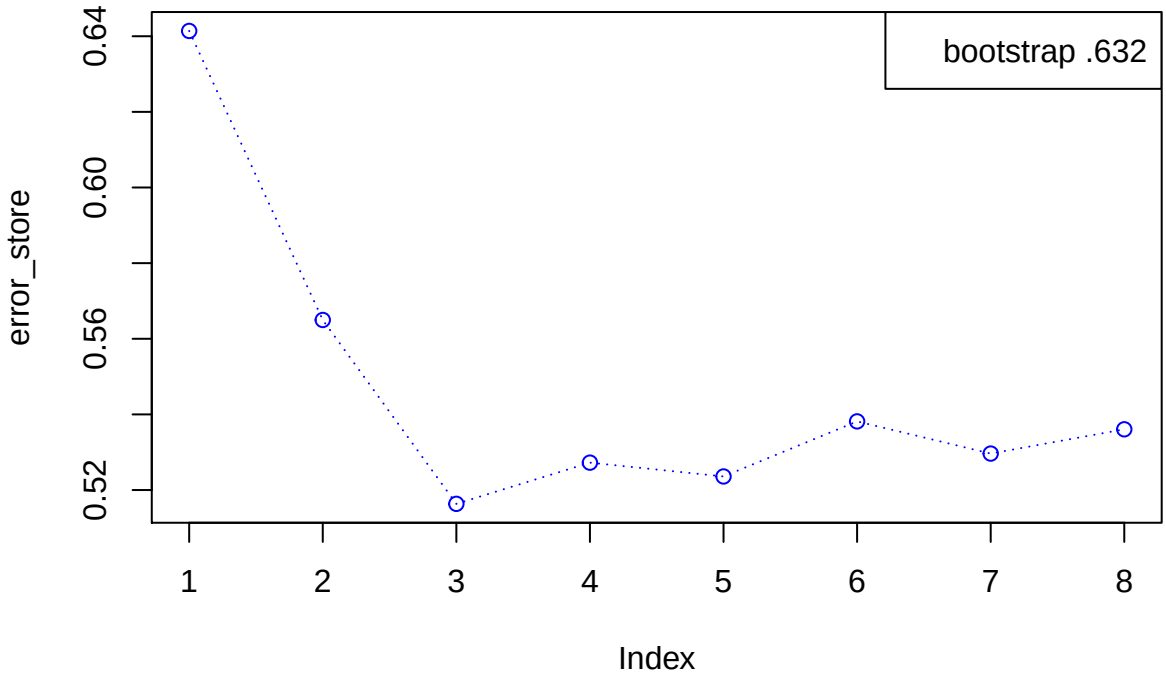
```
## [1] 0.6752674 0.6001678 0.5810431 0.5869649 0.5706711 0.5872400 0.5578341
## [8] 0.5660369
```

```
## [1] 0.8217466 0.7747050 0.7622618 0.7661363 0.7554278 0.7663159 0.7468829
## [8] 0.7523543
```

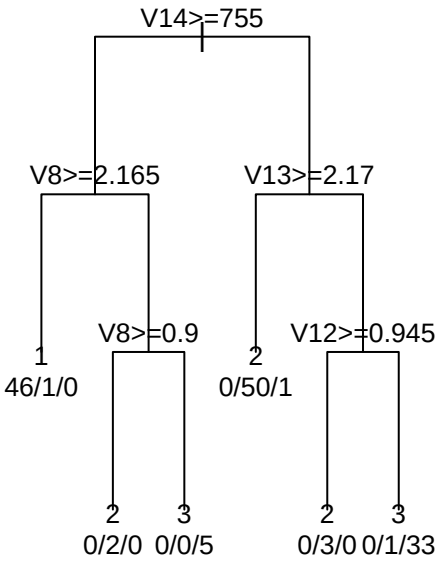
```
## [1] 7
```

For bootstrap method, when we are sampling data from the priginal data, 3th sampling data's error is minimum(about0.513) based on the bootstrap plot.

```
## [1] 0.6414038 0.5649694 0.5163469 0.5272425 0.5235855 0.5381463 0.5296344
## [8] 0.5360561
```



2) Based on the classification tree, it classifies the data of V1 into 46 class "1", 55 class "2" and 38 class "3". It seems that V14 is the most important factor in determining V1.With this, V8, V13 and V12 are important factors in determining V1. For example, class"1" is greater than 755 of Proline(V14). It is also greater than 2.65 of Nonflavanoid phenols(V8).



Train error is about 0.655. Test error is 0.139. and the error of pruned is 0.139. Wine data has only 13 variables so it genrates a small tree. Usually a smaller tree with fewer splits might lead to lower variance and better interpretation at the cost of little bias. That's why test error and the error of pruned are same. With this, training samples fall into each node is 142 and 36 for testing samples.

```
##
## pred_train_wine  1  2  3
##               1 46  1  0
##               2  0 55  1
##               3  0  1 38
```

```
## Warning in `!=.default`(pred_train_wine, test_wine$V1): longer object
## length is not a multiple of shorter object length
```

```
## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
## [1] 0.6549296
```

```
##
## pred_test_wine  1  2  3
##              1 11  1  0
##              2  2 12  1
##              3  0  1  8
```

```
## [1] 0.1388889
```

```
##           CP nsplit  rel error    xerror       xstd
## 1 0.50588235      0 1.00000000 1.0000000 0.06872010
## 2 0.34117647      1 0.49411765 0.5764706 0.06664636
## 3 0.05882353      2 0.15294118 0.2705882 0.05165047
## 4 0.03529412      3 0.09411765 0.2470588 0.04976674
## 5 0.02352941      4 0.05882353 0.2117647 0.04664273
## 6 0.00000000      5 0.03529412 0.1176471 0.03586938
```

```
##
## pred_train_wine_pruned  1  2  3
##                       1 11  1  0
##                       2  2 12  1
##                       3  0  1  8
```

```
## [1] 0.1388889
```

```
## [1] 142
```

```
## [1] 36
```

I chose the "Smarket"dataset. "Smarket"data is about daily percentage returns for the S&P 500 stock index between 2001 and 2005. First, I removed the column "Today" and "Year" because, I want to predict "Direction" using Lag1~5 and Volume. Then, I generated logistic regression model using glm function. The test error of logistic regression is 0.512. In order to make a prediction as to whether the market will go up or down on a particular day, I converted these predicted probabilites into class labels 0 for "down" and 1 for"up". And then I generated the table(confusion matrix) that indicate correct prediction, while the off-diagonals represent incorrect predictions. In my case, I think that do committee machine is bettr then non-ensemble method. Ensemble method gives a better prediction in oter words, it gives me a more exact prediction than the non-ensemble method. With this, it give me a more stable model than non-ensemble method. It means that the model of ensemble method is less noisy than the model of non-ensemble method. The aggregate opinion of a multiple models is less noisy than other models. So it allows me to predit the model more precisely.

```
##    logit.pred
##      0  1
##   0 38 83
##   1 45 84
```

```
## [1] 0.512
```

For begging, the test error is 0.468.

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
##    bagging.pred
##      0  1
##   0 52 69
##   1 48 81
```

```
## [1] 0.468
```

For boosting, the test error is 0.532.

```
##    boosting.pred
##      0  1
##   0 72 49
##   1 84 45
```

```
## [1] 0.532
```

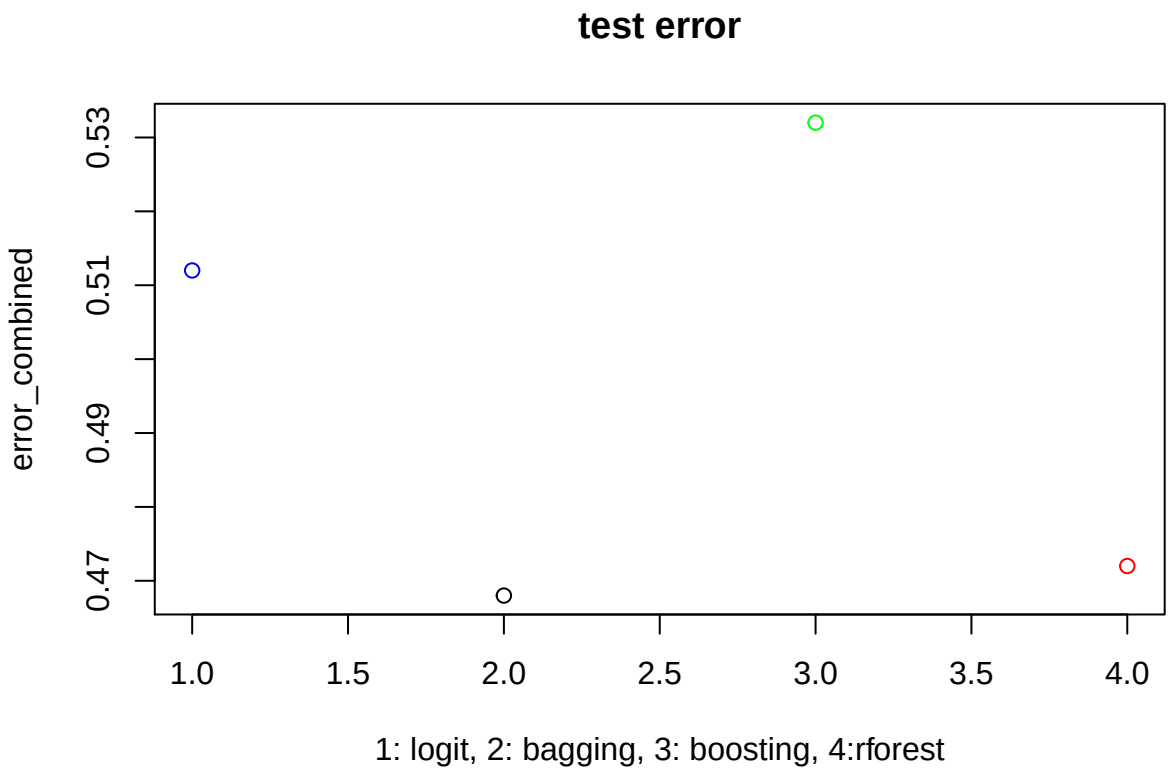
For random forest, the test error is 0.532

```
## Warning in randomForest.default(m, y, ...): The response has five or fewer
## unique values. Are you sure you want to do regression?
```

```
##    rforest.pred
##      0  1
##   0 49 72
##   1 46 83
```

```
## [1] 0.472
```

Based on the plot, test error fo bagging is the minimum among 4 errors.


test error
error_combined
0.47
0.49
0.51
0.53
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1: logit, 2: bagging, 3: boosting, 4:rforest