EAS 503-Homework 4

Boobalaganesh Ezhilan
UB Number - 50288429
November 25, 2018

Problem 1

Knowing the Problem
In this problem, we predict the number of patients who have prostate cancer by carrying out a best-subset linear regression analysis and finding the AIC, BIC, five- and tenfold cross-validation, and bootstrap .632 estimates of prediction error.

Splitting the Data
The dataset has 97 rows, out of which I had taken 50 percent of the training data set and other 50 percent of data as test data set. So training data set has 49 observations and test set has 50 observations

Fitting the Linear Regression Model
The mean square loss or MSE when using Least Square Method for predicting the number of application is given by 28740.8.
The Mean Square Error and Root Mean Square Error for least square model are extremely large in this case.

Often in multiple regressions, many variables are not associated with the response. Irrelevant variables lead to unnecessary complexity in the resulting model. By removing them (setting coefficient = 0) we obtain a more easily interpretable model. However, using OLS makes it very unlikely that the coefficients will be exactly zero. Here we explore some approaches for automatically excluding features using this idea.

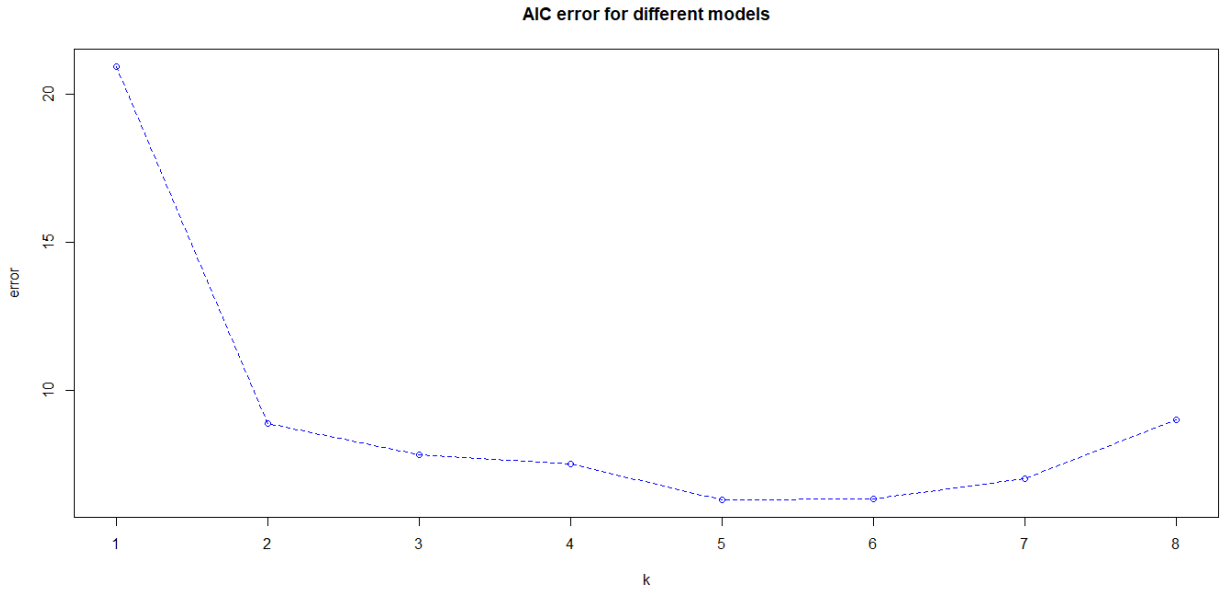
Performing the Best subset selection
I had performed the best Subset selection for the training data with nvmax=9 and plotted against BIC and C_p. The model with the least C_p error is for 5 and BIC it is 2.In these case no of the variable where the C_p is minimum is different from no of the variable where the BIC is minimum.
AIC criterion is defined for a large class of models fit by maximum likelihood (ML). In the case of a Gaussian model, ML and OLS are equivalent. Thus for OLS models, AIC and C_p are proportional to each other and only differ in that AIC has an additive constant term.

BIC is derived from a Bayesian point of view but looks similar to AIC and C_p. For an OLS model with d predictors, the BIC replaces the 2 d σ^2 from C_p with \(log (n) d σ^2\), where n is the number of observations. Since log n > 2 for n > 7, the BIC statistic generally places a heavier penalty on models with many variables and results in smaller models

In our case, as the response variable follows the Gaussian distribution, So the variance is one in these case so the AIC value will become equal to C_p.

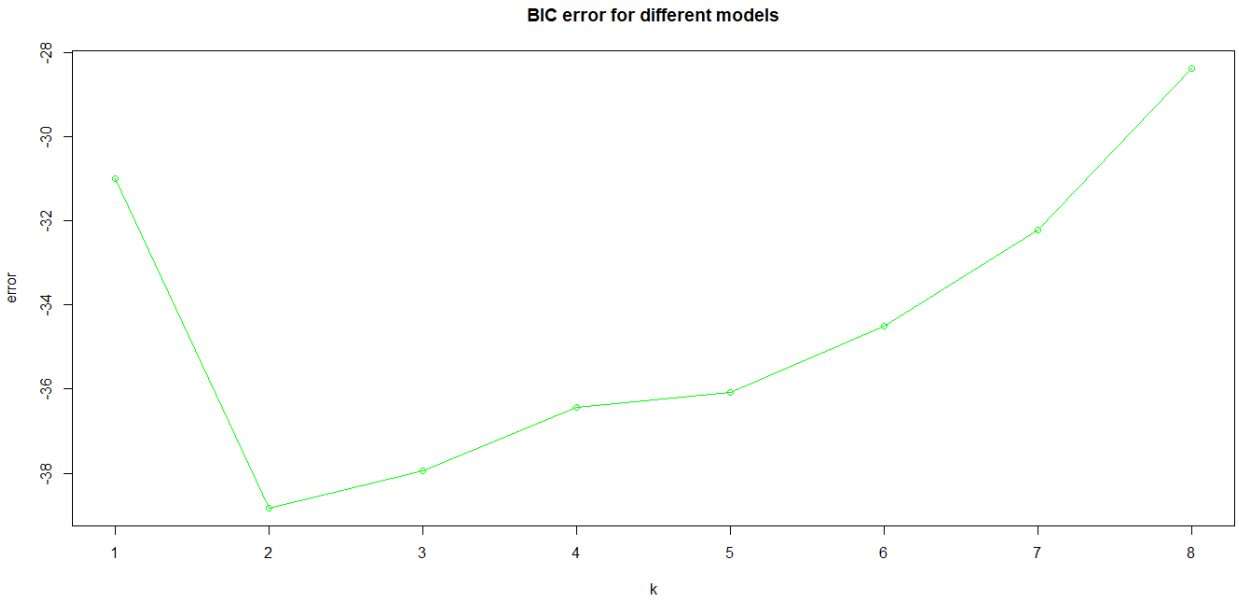
From the below picture we know that AIC is minimum for the 5_{th} variable (i.e. 6.321) and from the plot also we know that AIC error is minimum for 5_{th} variable

```
> fwd_sum$cp
[1] 20.935773  8.900191  7.825827  7.534761  6.321346  6.333199  7.030886  9.000000
```



From the below picture we know that AIC is minimum for the 2_{th} variable (i.e. -38.82493) and from the plot also we know that AIC error is minimum for 2_{nd} variable

```
> fwd_sum$bic
[1] -31.00659 -38.82493 -37.94417 -36.42928 -36.07093 -34.50933 -32.21357 -28.38036
```



Five – and tenfold cross-validation

We will now consider how to do this using the validation set and cross-validation approaches. In order for these approaches to yield accurate estimates of the test error, we must use only the training observations to perform all aspects of model-fitting, including variable selection. Therefore, the determination of which model of a given size is best must be made using only the training observations.

We now try to choose among the models of different sizes using cross-validation. This approach is somewhat involved, as we must perform the best subset selection within each of the k training sets.

In order to do that, first, we create a vector that allocates each observation to one of k=10 folds and we create a matrix in which we will store the results

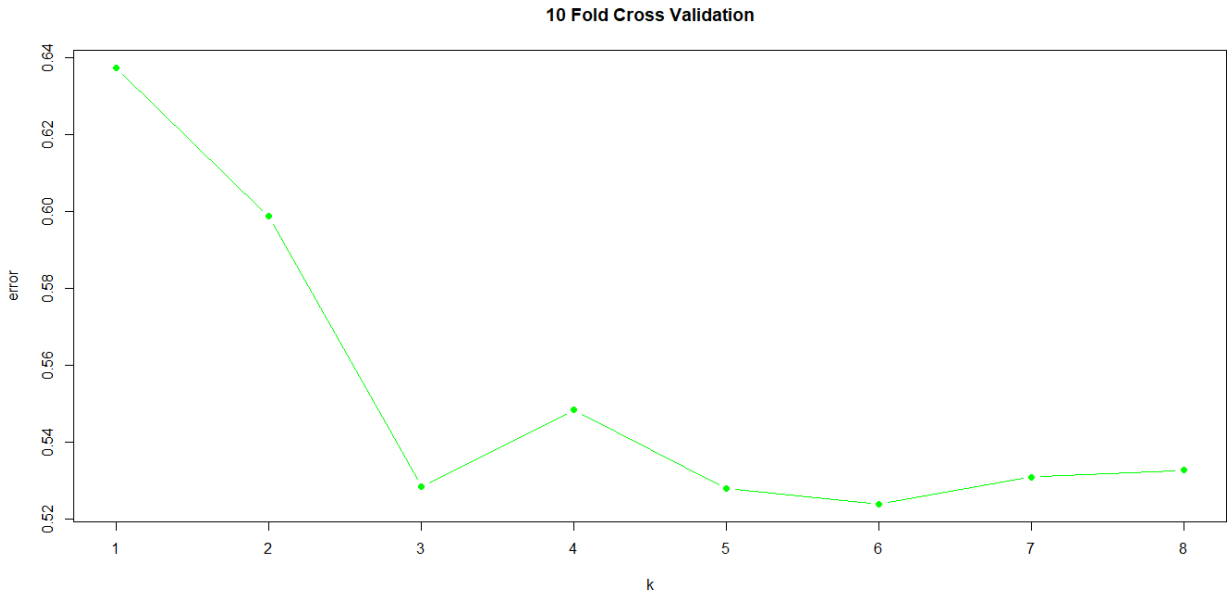
Now we write for loop that performs cross-validation. In the j_{th} fold, the elements of folds that equal j are in the test set, and the remainders are in the training set. We make our predictions for each model size compute the test errors on the appropriate subset, and store them in the appropriate slot in the matrix cv.error( ).

This has given us a 10× 9 matrix, of which the $(i, j)_{th}$ element corresponds to the test MSE for the I_{th} cross-validation fold for the best j-variable model. We use the apply function to average over the columns of this matrix in order to obtain a vector for which the j_{th} element is the cross-validation error for the j_{th} variable model.

We see that 10-fold cross-validation selects a 6-variable model.

The below picture is the 10-fold cross-validation error for the best subset model

```
        1         2         3         4         5         6         7         8
0.6372931 0.5986745 0.5283695 0.5482465 0.5277627 0.5238233 0.5308108 0.5326298
```

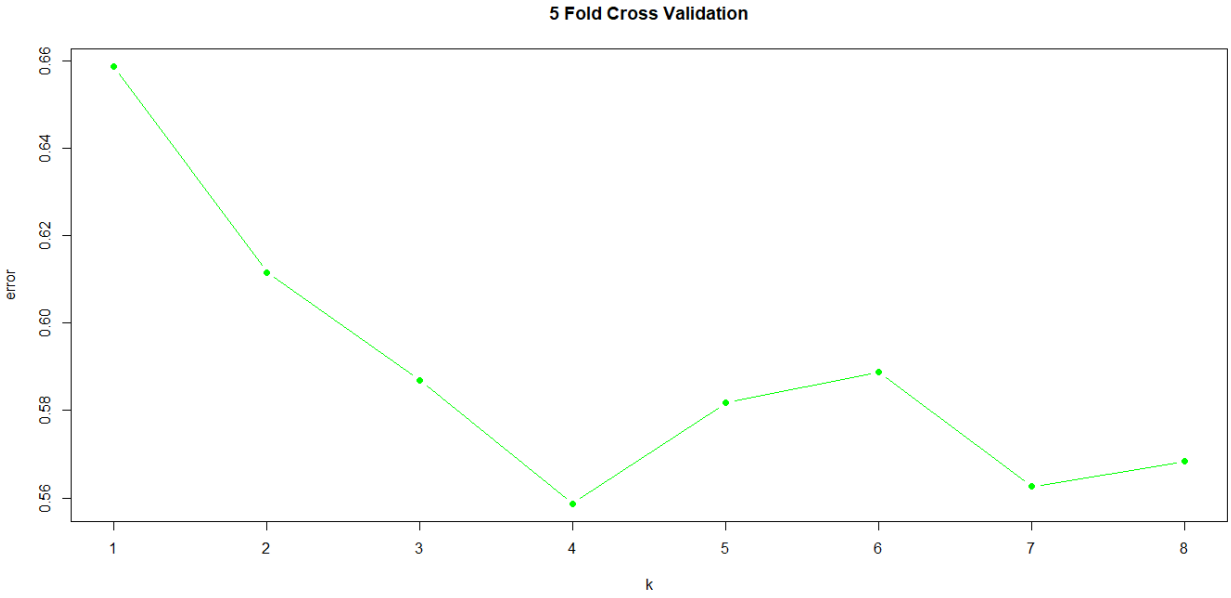


Similarly for the 5-fold cross-validation, The 5× 9 matrix, of which the $(i, j)_{th}$ element corresponds to the test MSE for the i_{th} cross-validation fold for the best j-variable model. We use the apply function to average over the columns of this matrix in order to obtain a vector for which the j_{th} element is the cross-validation error for the j_{th} variable model.

We see that 10-fold cross-validation selects a 6-variable model.

The below picture is the 5-fold cross-validation error for the best subset model

```
        1         2         3         4         5         6         7         8
0.6586470 0.6115945 0.5869321 0.5586724 0.5817036 0.5888256 0.5626422 0.5683451
```



Bootstrap .632

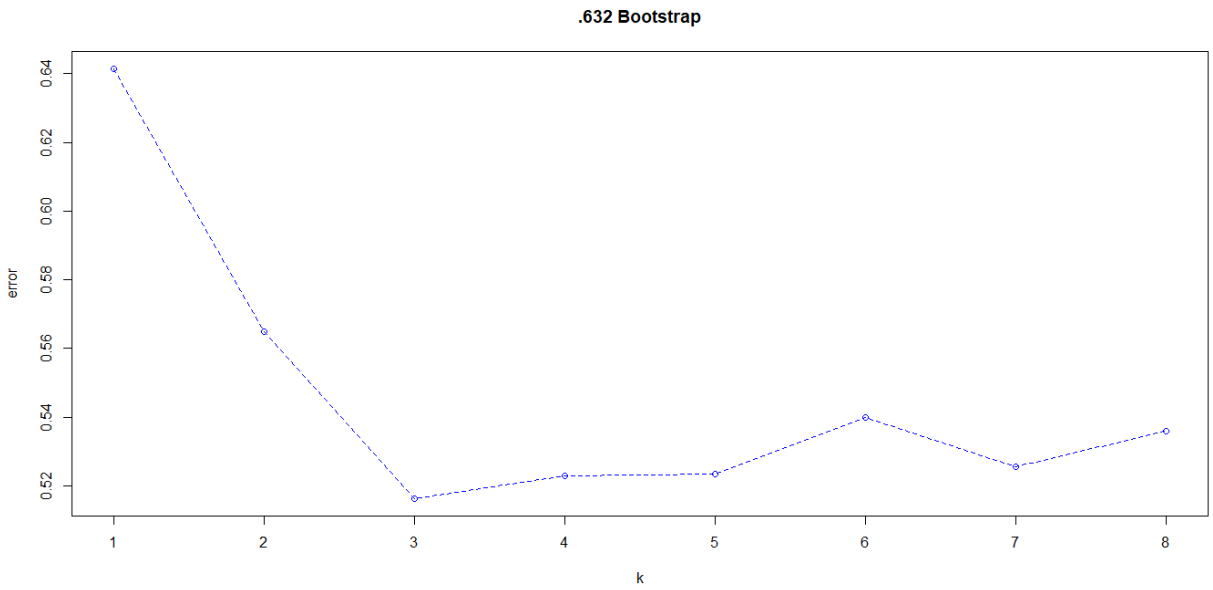
A bootstrap is a general tool for assessing statistical accuracy. As with cross-validation, the bootstrap
Seeks to estimate the conditional error ErrT , but typically estimates well only the expected prediction error Err. The basic idea is to randomly draw datasets with replacement from the training data, each
Sample the same size as the original training set.

When we apply .632 to the prostate data we get that bootstrap chooses 3 variable models. The below picture is the 5-fold cross-validation error for the best subset model

```
> error_store
[1] 0.6414038 0.5649694 0.5163469 0.5229835 0.5235855 0.5400557 0.5257113 0.5360561
```

Conclusion

In these particular problems and fitting methods, minimization of either AIC, cross-validation or bootstrap yields a model fairly close to the best available. Note that for the purpose of model selection,
Any of the measures could be biased and it wouldn't affect things, as long as the bias did not change the relative performance of the methods.
The error estimated using AIC and BIC overestimates the actual error and thus is only preferred when we have less data points because we would want to calculate for all our data into our training model. But if we have enough data points we would want to have our test set and use it to get an estimate of our error. But we may again get a wrong estimate of our error if we have an unfortunate split. So we prefer cross-validation methods to eliminate variance and we get an almost realistic error estimates
For example, the addition of a constant to any of the measures would not change the resulting chosen model. However, for many adaptive, nonlinear techniques (like trees), estimation of the effective number of parameters is very difficult. This makes methods like AIC impractical and leaves us with
Cross-validation or bootstrap as the methods of choice.
Thus estimation of test error for a particular training set is not easy in general, given just the data from that same training set. Instead, cross-validation and related methods may provide reasonable estimates of the prediction error Err.



Problem 2

Knowing the problem

In this problem, we need to classify the wine data into one of the following categories which are Barolo, Grignolino, Barbera. And we should provide appropriate –size classification tree for this dataset. And

We need to predict how many training and testing samples fall into each node.

Splitting the Data

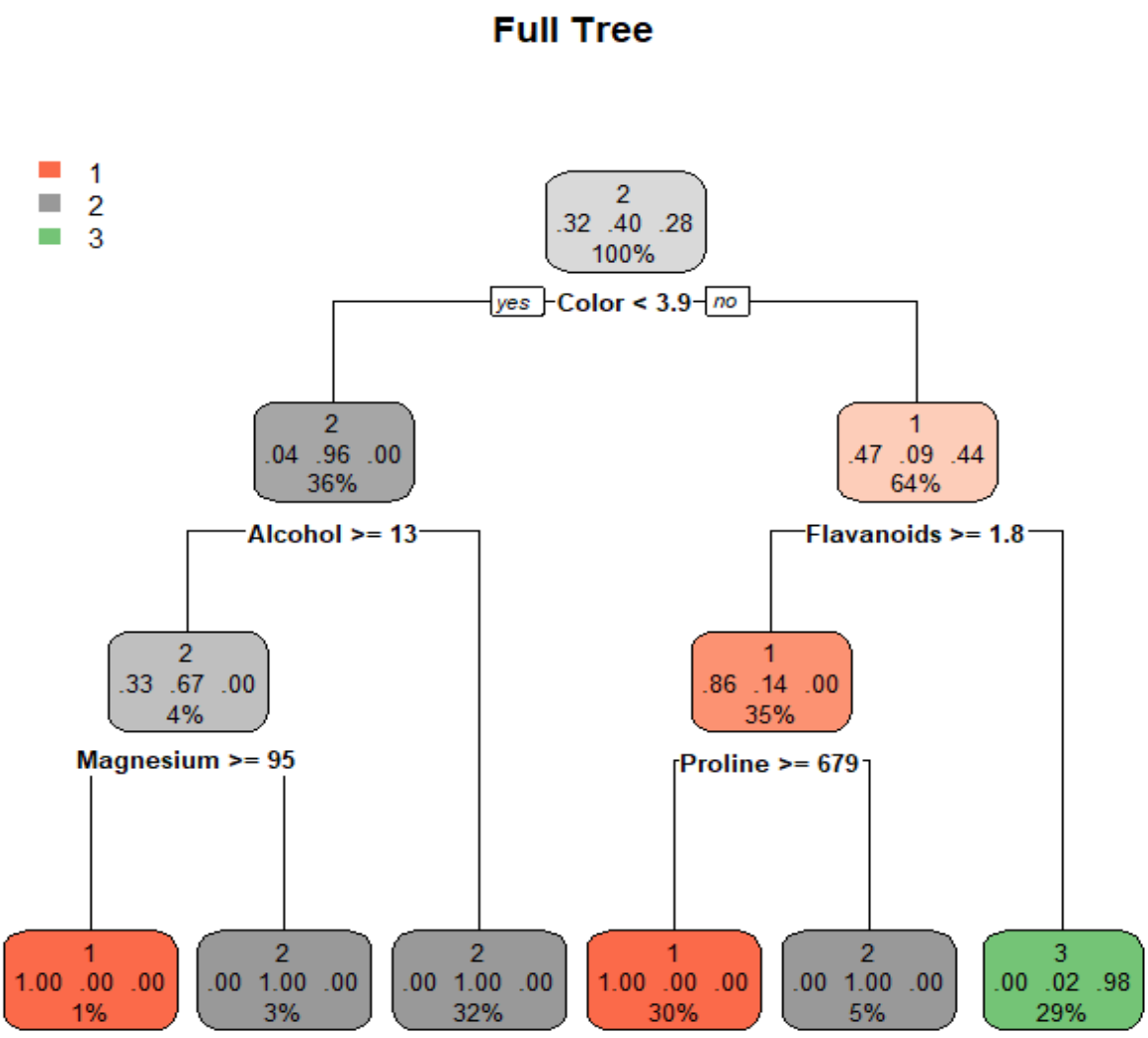
The dataset has 178 rows with 14 columns, out of which I had taken 80 percent of the data as training data set and remaining 20 percent of data as test data. So training data set has 142 observations and test set has 36 observations.

Decision Tree

A classification tree is very similar to a regression tree, except that it is used to predict a qualitative response rather than a quantitative one.

The predicted response for an observation is given by the mean response of the training observations that belong to the same terminal node. In contrast, for a classification tree, you predict that each observation belongs to the most commonly occurring class of training observations in the region to which it belongs. In interpreting the results of a classification tree, you are often interested not only in the class prediction corresponding to a particular terminal node region but also in the class proportions among the training observations that fall into that region

The decision tree build by the rpart() model can be visualized clearly from the below picture.



The number under each node indicates that the number of members of each class in that node. For example in the first node the value. So, the label "44 / 58 / 40" tells us that there are 44 cases of category 1 (Barabera, I infer), 58 cases of category 2 (Barolo), and 40 of category 3 (Grignolino).

From the summary of the wine model, we infer that the most important variable is v8 which has the score of 18 and next to important variables are v7,v11,v12,v13 which has the score of 12 and 11 which are almost similar.

Description of resulting tree

Following orange path from the root node

1. At the top, it is the overall probability of each wine into particular category. It shows the proportion of wine into different category.(.i.e. probability of particular wine going into category 1 is 0.32 and 0.40,0.27 for category 2 and category 3).

2. This node asks whether the color of the sample is greater than 1.2. If yes, then you go down to the root's left child node (depth 2). 0.44 percent goes to category 1 and 0.53,0.03 goes into category 2 and category 3
3. In the second node, you ask if the wine sample has v18 value greater than 755.if yes, then you go down to the left child node of the second node from the root. 0.96 percent goes to category 1 and 0.04,0.00 goes into category 2 and category 3
4. You keep on going like that to understand what features impact the likelihood of category of wine.

One of the many qualities of Decision Trees is that they require very little data preparation. In particular, they don't require feature scaling or centering.
By default, rpart() function uses the Gini impurity measure to split the note. The higher the Gini coefficient, the more different instances within the node.
When we apply predict function into model we can find the number of wine samples goes into which category. From the below table we can find which sample goes into which category

```
> table_mat
   tree_pred
     1  2  3
  1  7  0  0
  2  1 18  0
  3  0  1  9
> |
```

We can infer from the above figure that one of the categories 1 wine sample goes into category 2 and one of the category 2 wine samples goes into category 3 wine sample. We can do the sample by using confusion matrix also.

Appropriate size for classification

Based upon the pruned tree we came to the conclusion that before pruning the tree I had a test error a test error of about 0.1388889 after pruning the tree error rate came around 0.1111 ,so before pruning I had taken min split as 5 but my C_p value is minimum for the number of nodes equals 4 so taken 4 as my minimum split in pruning and calculated the error so conclude from my error rate that appropriate size for my tree is 4

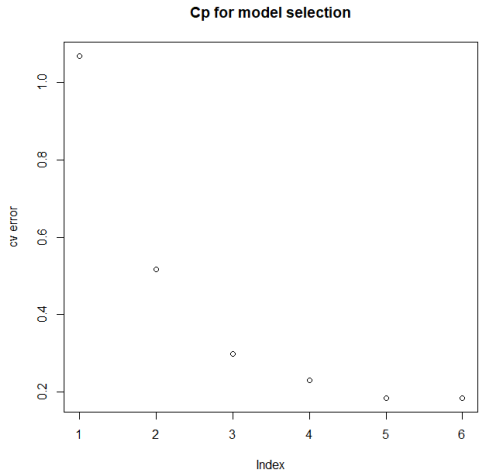
Training Sample fall into which node?

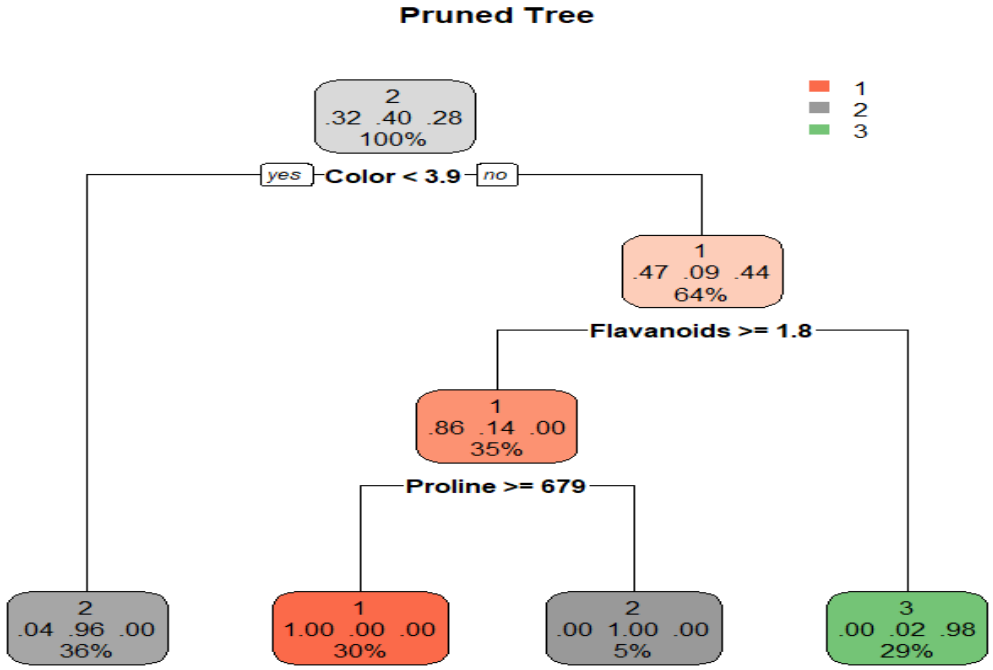
From the leaf node of the tree infer that we have 6 leaves nodes first node have 1 percent of data (.i.e. 2 data points falls into that leaf node) which is category 1 , second leaf node have 3 percent of the data (.i.e. 5 data points falls into that leaf node) which is of category 2,third leaf node have 32 percent of the data(.i.e. 56 data points falls into that leaf node) which is of category 2,Fourth leaf node have 30 percent of the data(.i.e. 56 data points falls into that leaf node) which is of category 1,Fifth leaf node have 5 percent of the data(.i.e. 9 data points falls into that leaf node) which is of category 2,sixth leaf node have 29 percent of the data(.i.e. 51 data points falls into that leaf node) which is of category 3.

Test sample fall into which node?

In order to get the leaf info for test data, I used to run predict() with type="matrix" and infer it. This returns, confusingly, a matrix produced by concatenating the predicted class, the class counts at that node in the fitted tree, and the class probabilities

```
> unique(testFeatures[,2:4])
    NofClass1onNode NofClass2onNode NofClass3onNode
7                48               1               0
67                0               0              26
70                0              47               0
71                0               1               6
72                0               3               0
147               0               2               1
>
```





Which row in which node?

From the below table we can infer that which row falls into which leaf node. For example consider first

Value which represents 34th which falls into 9th node similarly second value is 104 which falls into 6th node likewise we calculate which row falls into which node

```
> wine_model$where
 34 104 127 151 174 129  81  25 121  14  42  75 155 133 152 131   5   7 134  62 173  98  31 109 168  16
  9   6  10  11  11   6   6   4   6   9   9   6  11  11  11  11   9   9  11  11  11   6   9   6  11   9
172  28  10  90  66  88 143  21 161 157 156 105   3  76  18 154 149  83  36 166 119 147  58 142  22 103
 11   9   9   6  10   6  11   9  11  11  11   6   9   6   9  11  11   6   9  11   6  11   9  11   9   6
132  91 171 128  50 163  27 120 110 177 170 116   9  70 111  44   2  77 135 136  80  93 107  48  99  78
 11   6  11   6   9  11   9   6   6  11  11   6   9   6   6   9   9  10  11  11   6   6   6   9  10   6
 86  60 125 124  87   6  67 150  74  24  49  53 176  54  92 106  45 153  37 144  40 146 100 138  11   4
  6   6   6   5   6   9  10  11   6   9   9   9  11   9   6   6   9  11   9  11   9  11   6  11   9   9
167  26  41  51 165 126 118 139  96  17  68 158  43  15  73  38 160  85 164  61  63  59 101  52  95  20
 11   4   9   9  11   6   6  11   6   9  10  11   9   9   5   9  11   6  11   6   5   9   6   9   6   9
 65  55  72  35 112  64 145  13 178 114 130  71
  6   9   5   9   6  10  11   9  11   6   6   6
>
```

Problem 3

In the given problem we need to predict whether the given suburb is greater than or smaller than the Median of crime rate by applying classification algorithms like Logistic Regression, K-nearest neighbor and committee machines like bagging, boosting and random forest

Knowing the data

Using the Boston data set which is in the ISLR package with 506 observation and 15 variables we need to split the data into train data and testing data. Here in this problem, I had split the data into 60 percent of training data and 40 percent of test data. Now, my test data will be having 333 rows and test data will have 173 observations

Creating Response Variable

Based upon the median value of crime rate we will find the response for the given data. Which is 1 in case of the value of crime data is greater than the median value and 0 when crime data value is lesser than its median value. Generally, it is categorical variable with values 1 and 0.

Performing Logistic Regression

We will try to apply logistic regression over the training dataset and try to find the training error and test error. Just because the response variable is not of factor variable we will convert into factor variable and perform logistic regression.

When we look into the summary of the fit we will be getting significance stars .under the estimate in the second row is the coefficient associated with the variable listed to the left. The Standard error in the next column associated with these estimates. Next column would be normalized error value (i.e. dividing estimates with the standard error). By looking at the significance stars we can infer that nitrogen oxides concentration, index of accessibility to radial highways and the median value of owner-occupied homes in $1000s contributes the lot to the output response. Now we will calculate the Mean Square error for the logistic regression for both training data and test data set which is around 0.14 for the test data set

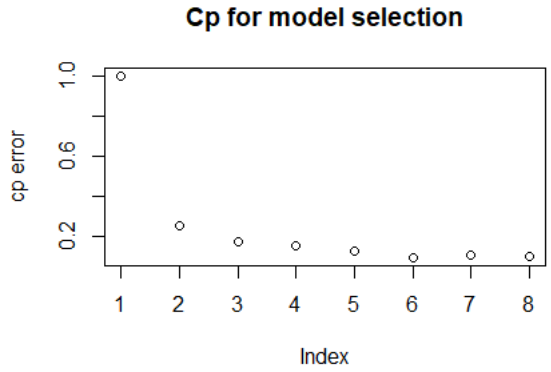
Training Error Logistic Error	Test Error Logistic Error
0.09309	0.132

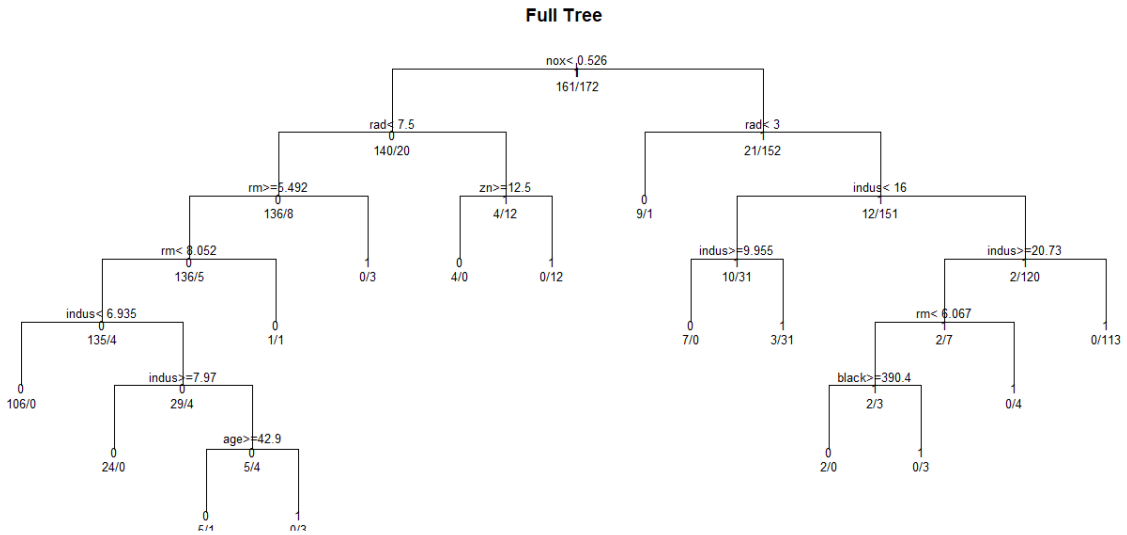
Applying K-nearest neighbor

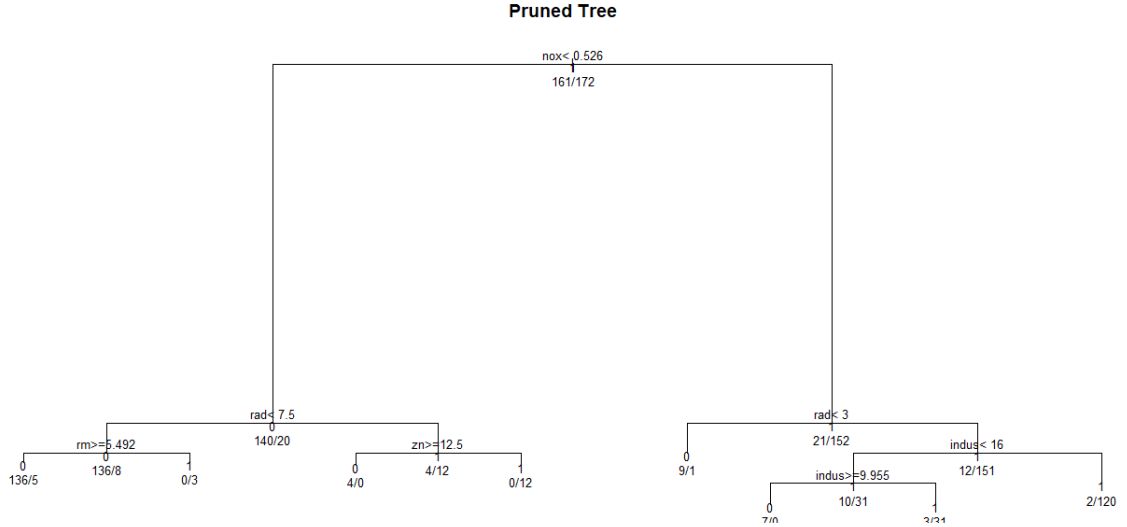
We will try to apply the Knn model to the training data and try to find the training error and test error. I had iterated the values of K from 1 to 10. The minimum test error is for K equals 2 and 3. And the test error is around 0.12716

Applying Decision Tree

When we applying the decision tree and look into the summary of the model the important variable according to decision trees are Indus and nox which have the importance score of about 18 for both of the predictors. While for others like tax and dis have a score of about 15 and 14.When we tried to predict the response variable mean error is about 0.0982 for the normal tree. When we tried to find the mean error for pruned dataset the error value is about 0.0693 which is better than a normal tree and way better than the K-nearest neighbor and Logistic regression method. The following graph is a relationship between tree nodes and cp error and it is minimum for 6 node model



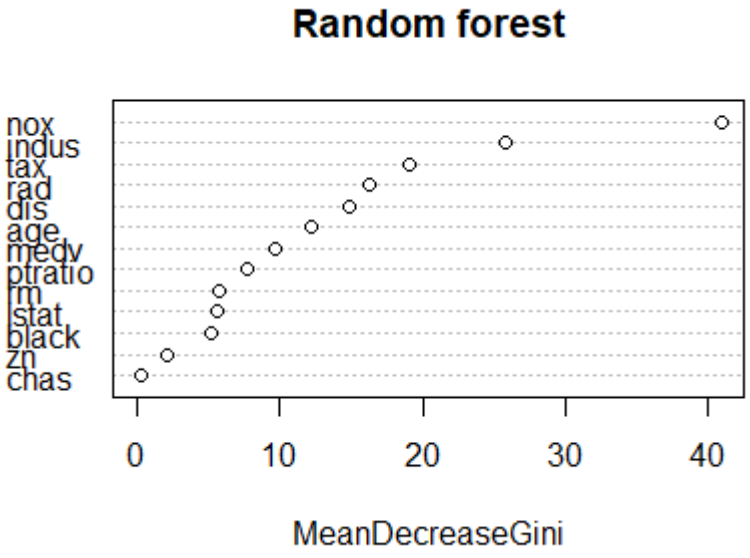




From the above tree structure, we can infer than a pruned tree has 7 nodes while the full tree has 14 nodes.

Applying Random Forest

When we apply random forest to the training dataset we infer that nox and Indus are the most important contributors to the response variable, which can also be seen from the graph.



This variable importance agrees with the summary of the decision tree. From the summary of the importance, we will be getting the variable importance which is given by

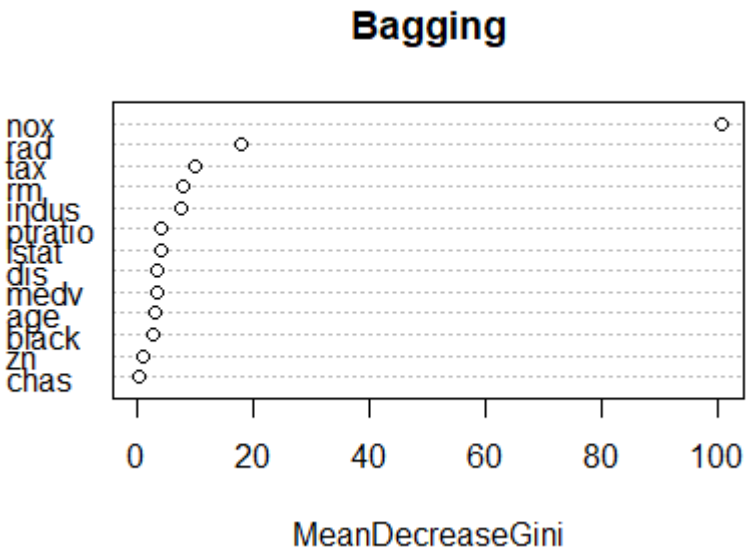
	MeanDecreaseGini
zn	1.962504
indus	26.296950
chas	0.291535
nox	36.305081
rm	5.922438
age	13.311518
dis	15.408693
rad	16.715930
tax	19.035875
ptratio	8.908122
black	4.983480
lstat	6.636555
medv	9.523499

>

From the above table, we infer that Indus which has the Gini index of about 26.2 play the important role in contributing to the response variables. When we apply the random forest to training dataset with a number of trees equals 10000 and tried to predict the response variable through the random variable we have a mean error of about 0.0809 which is better than logistic regression and K-nearest neighbor method but not efficient when compared to decision tree method.

Applying Bagging

Bagging is same as that of random forest, The fundamental difference between bagging and random forest is that in Random forests, only a subset of features are selected at random out of the total and the best split feature from the subset is used to split each node in a tree, unlike in bagging where all features are considered for splitting a node. In this problem I had taken mtry equals to the number of predictor variable which is 8



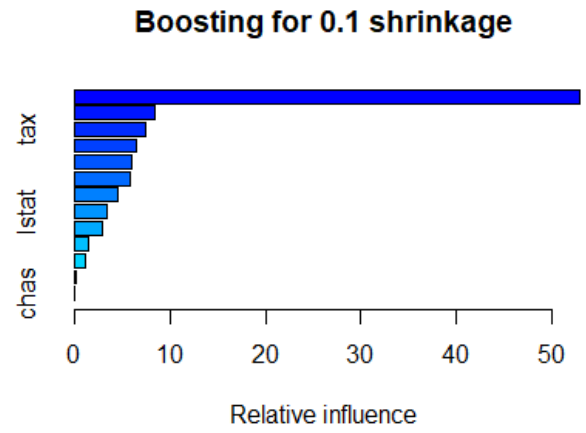
	MeanDecreaseGini
zn	1.1123736
indus	7.6886744
chas	0.1485782
nox	100.4526742
rm	7.7420471
age	2.9956281
dis	3.2630043
rad	17.8840591
tax	9.7258156
ptratio	4.4786171
black	2.9484420
lstat	4.1555712
medv	3.2953261

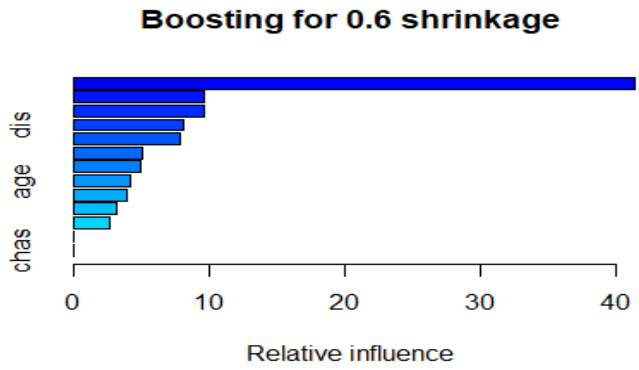
From the above table, we infer that Indus which has the Gini index of about 100.45 play the important role in contributing to the response variables which is very much higher than the value given by the random forest and decision tree. When we apply the random forest to training dataset with a number of trees equals 10000 and tried to predict the response variable through the random variable we have a mean error of about 0.0578 which is better than logistic regression and K-nearest neighbor method and also committee machine method like a random forest.

Applying Boosting

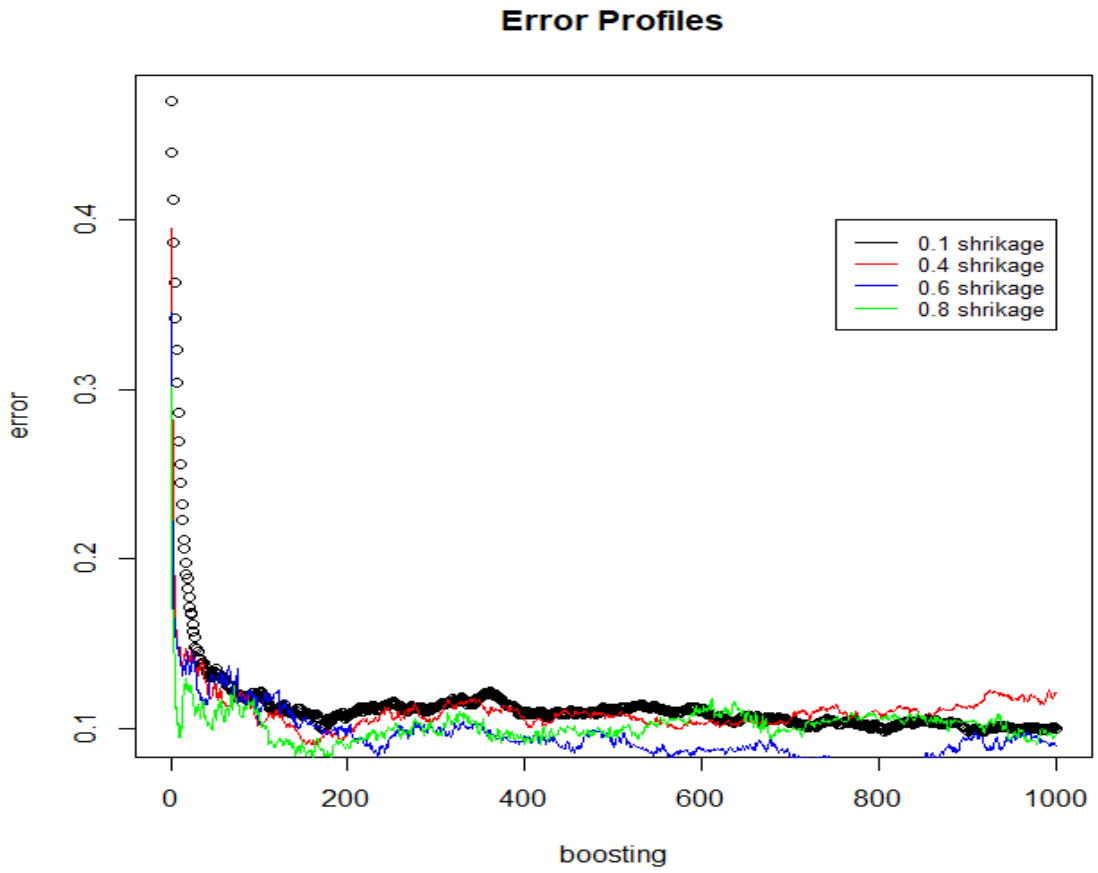
In these Boosting methods, I had tried to predict the response variable for two shrinkage factor which is 0.1 and 0.6. I had applied gbm() method to predict the response variable through boosting method.

The summary of the gbm() method gives the variable importance.





The above graph bars represent the importance of the variable importance. When we tried to apply boosting method to predict response variable we obtain mean error of about 0.1171 for shrinkage factor 0.1 and error of about 0.0923 for shrinkage factor 0.6.



The above graph represents the converging error for different shrinkage factor. From the graph, we can infer that the blue line is almost flat (i.e. blue line performs the best) the shrinkage coefficients corresponding to blue is 0.6.

```
shrinkage:0.1 shrinkage:0.4 shrinkage:0.6 shrinkage:0.8
   0.09949630    0.12110090    0.08958713    0.09663785
```

Advantages of committee method over other

The errors on test sets were: Logistic Regression = 0.132, KNN = 0.127, Random Forest = 0.080, Bagging = 0.0966, AdaBoost = 0.053.

From correlation plot between medv and other variable we infer that Random forest and Bagging performs better than the non committee methods like logistic regression and K-nearest neighbor.

This is because of the reason data points in the response variable are linearly inseparable which means that the data is unbalanced. Since non committee methods provides the better fit for the non-linear classifier. We can also use logistic regression even when it is less predictive because it is more interpretable or faster. Also, KNN does not perform as good as the committee machines methods because of high dimensional nature of our data. When we look into computational wise logistic regression is cheaper compared to the other. Random Forests

are used in practice to better generalize the fitment. RF provide a good balance between precision and over fitting.