CMDA 4654: Intermed Data Analytics & ML
Homework 4
Christopher J. Mobley
Due: Monday, 25APR16

1. (8.4 – 2) It is mentioned in Section 8.2.3 that boosting using depth-one trees (or stumps) leads to an *additive* model: that is, a model of the form

$$f(X) = \sum_{j=1}^{p} f_j(X_j)$$

Explain why this is the case. You can begin with (8.12) in Algorithm 8.2.

In boosting with depth-one trees (or stumps), the model is improved by fitting another depth-one tree to the residual of the previous tree and adding it to the model. Since each depth-one tree is a split with regards to one predictor, $X_j < t$, the final model with B different stumps can be grouped in regards to their predictor, for example grouping all $\hat{f}^b$ that involve predictor X_1, and added together forming the equation above, which is additive in regards to each predictor. The mathematically proof for this is found below

$$f(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

At step 0

$$\hat{f}(x) = 0 \text{ and } r_i = y_i$$

At step 1

$$\hat{f}^1(x) = c_1 \cdot I(X_1 < t_1) + c'_1$$

So,

$$\hat{f}(x) = \lambda \hat{f}^1(x) \text{ and } r_i = y_i - \lambda \hat{f}^1(x_i)$$

In order to maximize our fit to the residuals, we fit B distinct model where B = j

$$\hat{f}^j(x) = c_j \cdot I(x_j < t_j) + c'_j$$

So,

$$\hat{f}(x) = \lambda \hat{f}^1(x) + \cdots + \lambda \hat{f}^j(x)$$

and

$$r_i = y_i - \lambda \hat{f}^1(x) - \cdots - \lambda \hat{f}^j(x)$$

Since each iterations fit is based on a distinct predictor, there are only p distinct fits. So, after grouping the equation becomes.

$$\hat{f}(X) = \sum_{j=1}^{p} f_j(X_j)$$

2. (8.4 – 5) Suppose we produce ten bootstrapped samples from a data set containing red and green classes. We then apply a classification tree to each bootstrapped sample and, for a specific value of X, produce 10 estimates of P(Class is Red | X):

   0.1, 0.15, 0.2, 0.2, 0.55, 0.6, 0.6, 0.65, 0.7, and 0.75.

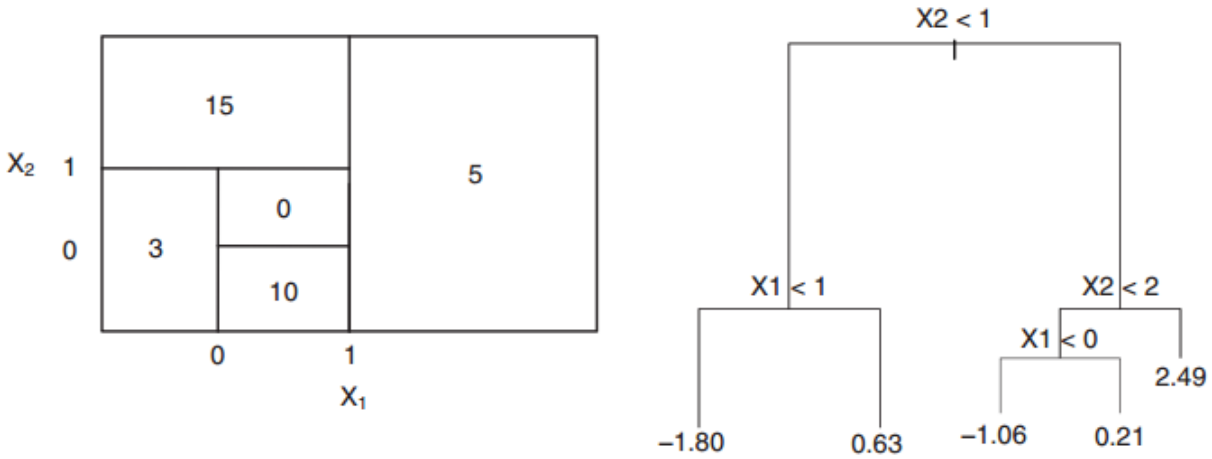

Figure 1. Left: A partition of the predictor space corresponding to Exercise 4a. Right: A tree corresponding to Exercise 4b.

There are two common ways to combine these results together into a single class prediction. One is the majority vote approach discussed in this chapter. The second approach is to classify based on the average probability. In this example, what is the final classification under each of these two approaches?

> With the majority vote approach, we would classify X as Red. This is due to the fact it is occurs mostly frequently, 6 out of 10 times. With the averaging probability approach, we would classify X as Green. This is due to the fact, that the average of the 10 probabilities given above is 0.445 and thus Green.

3. (8.4 – 8) In the lab, a classification tree was applied to the Carseats data set after converting Sales into a qualitative response variable. Now we will seek to predict Sales using regression trees and related approaches, treating the response as a quantitative variable.
   a. Split the data set into a training set and a test set.

b. Fit a regression tree to the training set. Plot the tree, and interpret the results. What test MSE do you obtain?

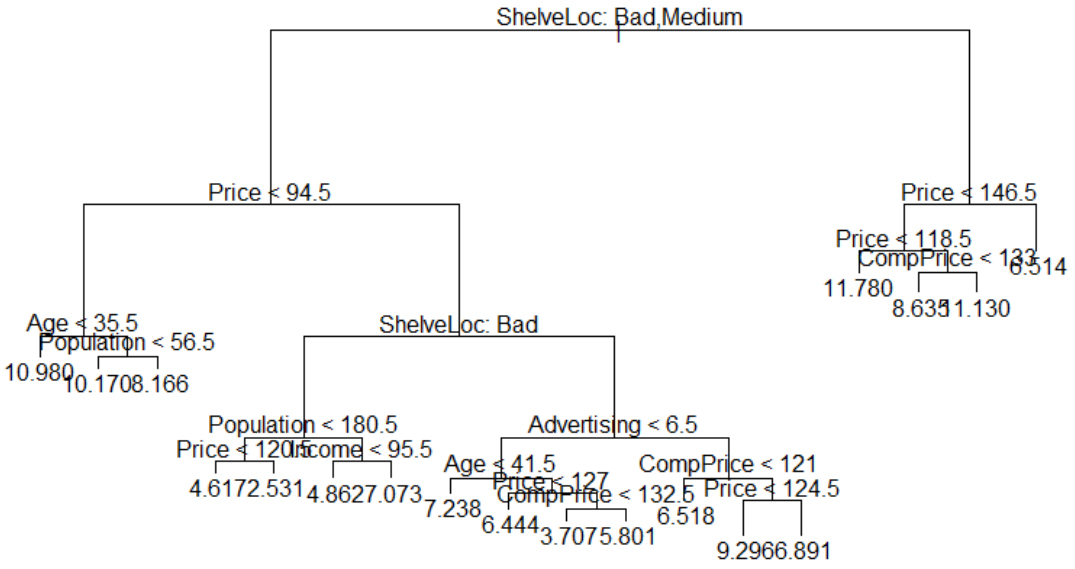

Figure 2: Initial Unpruned Carseats regression tree

```
Regression tree:
tree(formula = Sales ~ ., data = train)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Age"         "Population"  "Income"      "Advertising" "CompPrice"
Number of terminal nodes:  18
Residual mean deviance:  1.828 = 332.6 / 182
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-3.47100 -0.85760  0.01643  0.00000  0.96960  3.09800
```

Figure 3: Initial unpruned Carseats regression tree summary

Figures 2 and 3 show the layout and summary for the initial unpruned regression tree for the Carseats dataset. Only 7 of the 10 variables were used to make the tree with a depth of 18. The MSE of this initial unpruned regression tree is 4.477452.

c. Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test MSE?

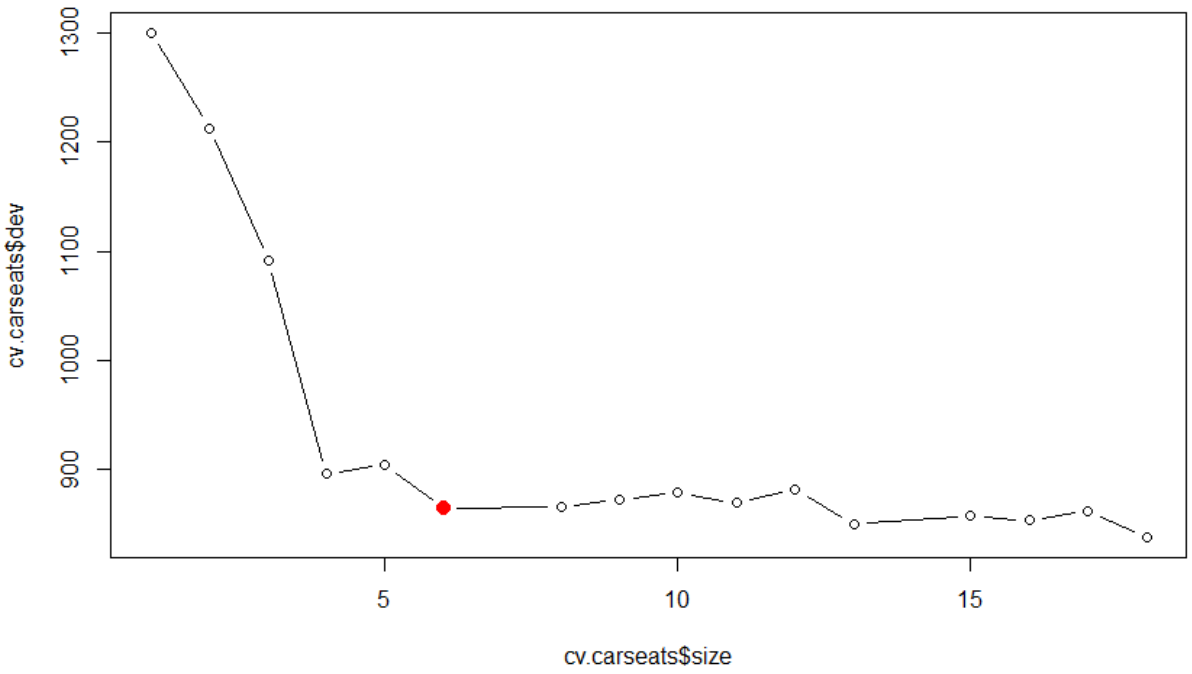

Figure 4: Cross Validation for Carseats regression tree model showing deviance versus size

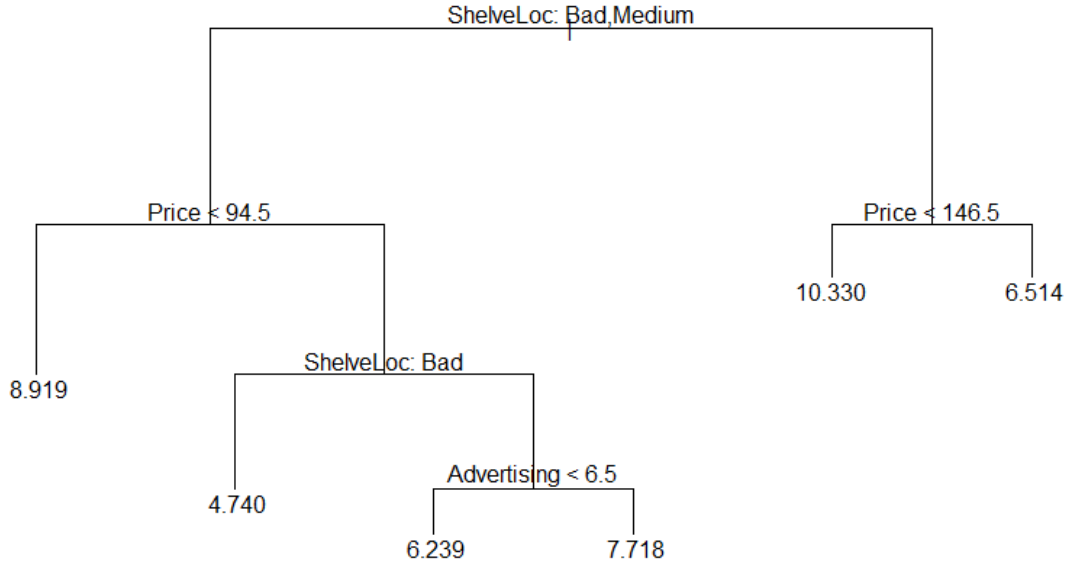

Figure 5: Pruned Carseats regression tree with size of 6

Figure 4 show that model deviance level off with a terminal size of approximately 6. Figure 5 shows the pruned regression tree for the Carseats dataset. The MSE for the pruned tree is 5.208115. This is slightly higher than that of the full regression tree show in part (b).

d. Use the bagging approach in order to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important.

	%IncMSE	IncNodePurity
CompPrice	21.4674924	127.767968
Income	0.9075028	60.629023
Advertising	11.9998791	86.644936
Population	-1.9984333	54.142939
Price	57.4299058	380.441111
ShelveLoc	49.7905463	348.452887
Age	14.5356992	118.623427
Education	2.7940569	42.414542
Urban	-0.6258907	7.397644
US	5.2112303	10.687375

Figure 6: Variable Importance for Carseats bagged regression tree

Figure 6 shows that the Price and ShelveLoc variables appear to be the most important variables in our bagged regression tree for the Carseats dataset. The MSE for the bagged regression tree is 2.932783, which is lower than the previous two model.

e. Use random forests to analyze this data. What test MSE do you obtain? Use the importance() function to determine which variables are most important. Describe the effect of m, the number of variables considered at each split, on the error rate obtained.

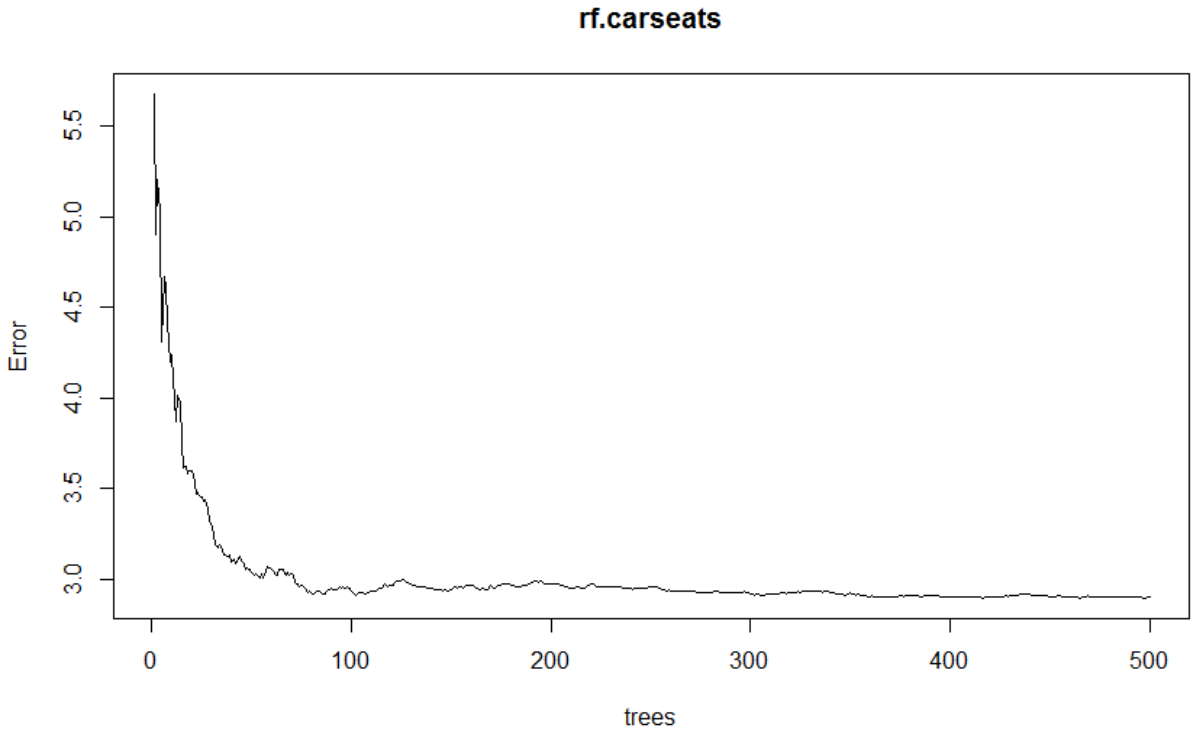

Figure 7: Random forest regression model error for a given number of trees

	%IncMSE	IncNodePurity
CompPrice	10.8896740	114.42913
Income	1.9668734	93.81426
Advertising	13.9004159	114.95596
Population	-3.3907326	90.58748
Price	36.7085556	298.84473
ShelveLoc	33.1071231	259.08665
Age	12.2675136	150.19842
Education	0.7890977	60.82180
Urban	-1.2095544	10.28038
US	2.1490545	15.16011

Figure 8: Variable Importance for Carseats random forest regression model

Figure 7 shows that our error rate plateaus at approximately 100 tree. Unlike a normal regression tree in which all predictors are considered at each split, a decision forest only considers a random sample of m predictors. Figure 8 shows that the Price and ShelveLoc variables again appear to be the most important variables in our random forest regression model for the Carseats dataset. The MSE for random forest regression model MSE is 3.673223, which is slightly higher than our bagged regression tree but lower than the previous two models.

4. (8.4 – 10) We now use boosting to predict Salary in the Hitters data set.
    a. Remove the observations for whom the salary information is unknown, and then log-transform the salaries.

b. Create a training set consisting of the first 200 observations, and a test set consisting of the remaining observations.
c. Perform boosting on the training set with 1,000 trees for a range of values of the shrinkage parameter λ. Produce a plot with different shrinkage values on the x-axis and the corresponding training set MSE on the y-axis.

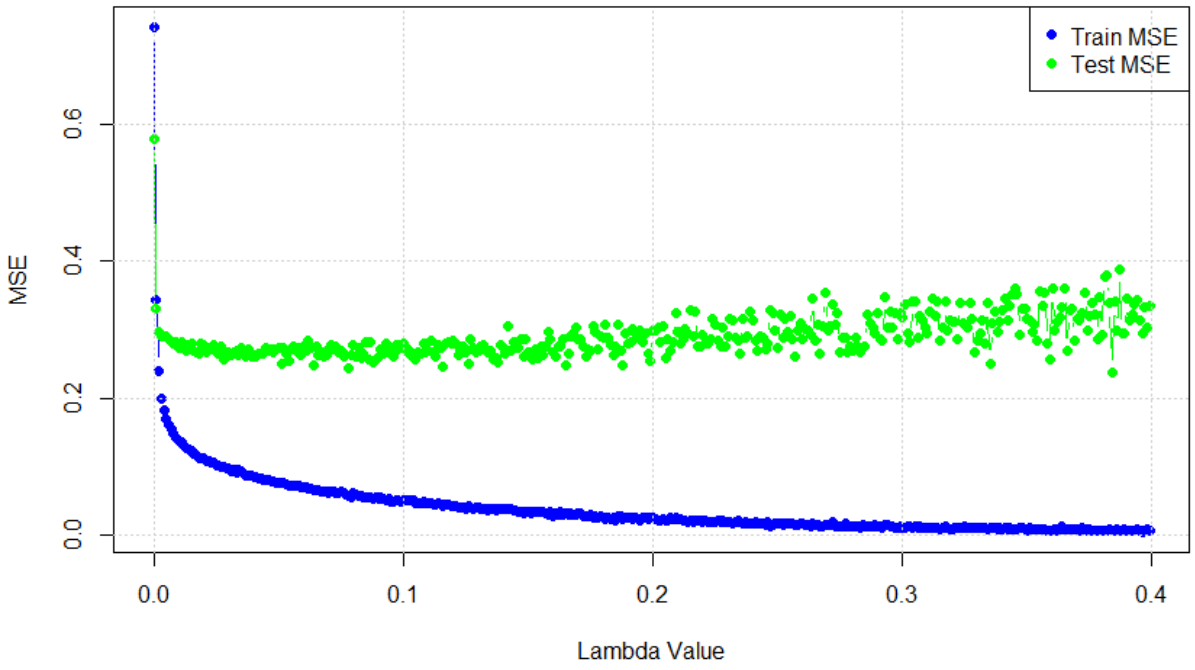

Figure 9: MSE for train and test of a boosted regression tree for the Hitter dataset

Figure 9 shows the MSE plotted versus Lambda for both test and train Hitter datasets by our boosted regression tree model.

d. Produce a plot with different shrinkage values on the x-axis and the corresponding test set MSE on the y-axis.

See (c)

e. Compare the test MSE of boosting to the test MSE that results from applying two of the regression approaches seen in Chapters 3 and 6.

The boosted regression tree's MSE is .2990196 is significantly lower than that of both the linear and lasso models, which have similar MSEs of .4917959 and .4709719 respectively.

f. Which variables appear to be the most important predictors in the boosted model?

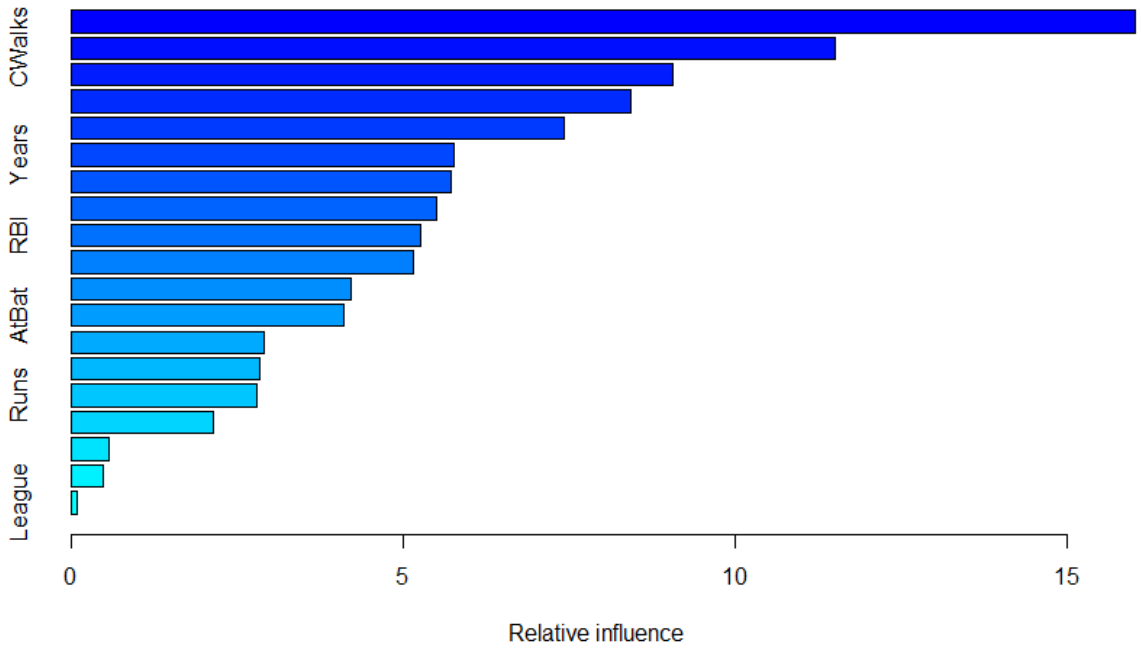

Figure 10: Summary of Relative Influence for the predictors of the boosted regression tree

	var	rel.inf
CAtBat	CAtBat	16.0213372
CWalks	CWalks	11.5074856
PutOuts	PutOuts	9.0534018
Walks	Walks	8.4206293
CHmRun	CHmRun	7.4138550
Years	Years	5.7583564
CRBI	CRBI	5.7161606
Assists	Assists	5.4966812
RBI	RBI	5.2614746
HmRun	HmRun	5.1610688
Hits	Hits	4.2189027
AtBat	AtBat	4.1117570
Errors	Errors	2.9078352
CHits	CHits	2.8538956
Runs	Runs	2.7947565
CRuns	CRuns	2.1457407
NewLeague	NewLeague	0.5678991
Division	Division	0.4872524
League	League	0.1015103

Figure 11: Summary of Relative Influence for the predictors of the boosted regression tree

Figures 10 and 11 show that the most important predictors are CAtBat, CWalks, and PutOuts.

g. Now apply bagging to the training set. What is the test set MSE for this approach?

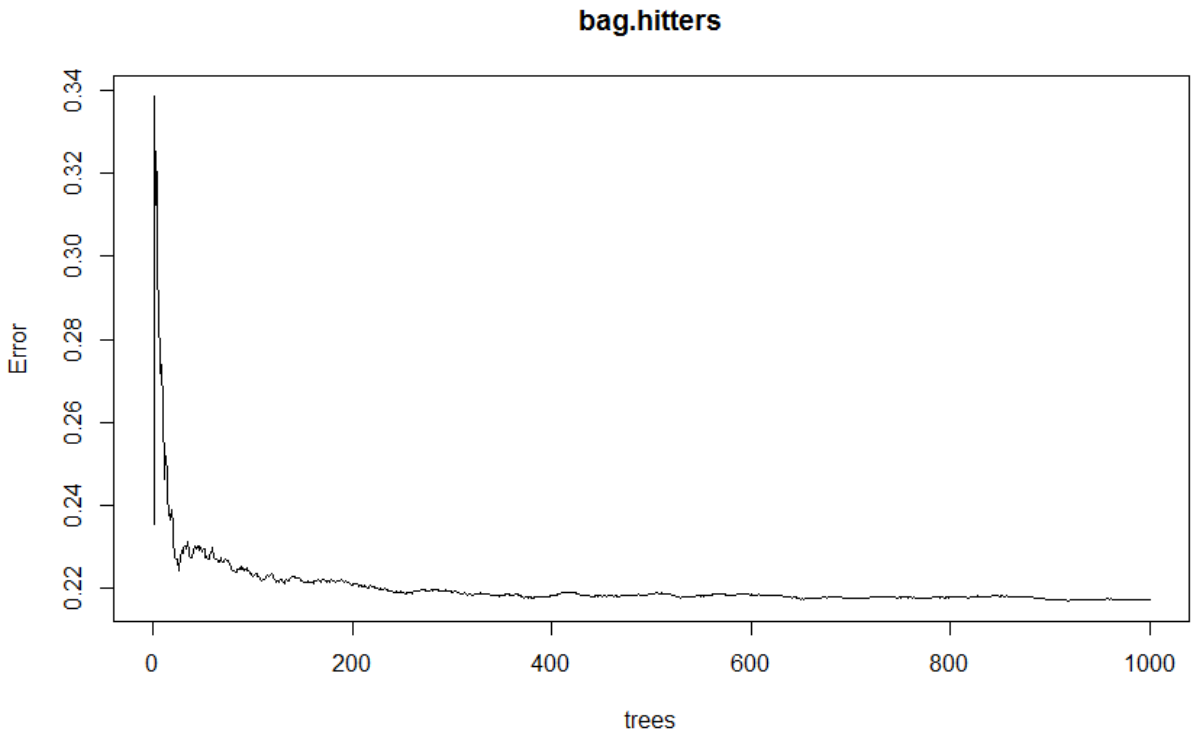

Figure 12: Bagged regression tree error for a given number of trees

	%IncMSE	IncNodePurity
AtBat	15.6912384	7.64599194
Hits	10.4076084	4.23141645
HmRun	2.2138684	1.47016327
Runs	7.1907957	2.93350034
RBI	3.0946531	3.21461015
Walks	13.4616005	7.19875386
Years	14.7128080	1.90730583
CAtBat	53.4793864	89.89191276
CHits	11.3101339	11.38565544
CHmRun	14.0046722	4.75584288
CRuns	16.7678837	10.55760256
CRBI	16.3043487	8.73782358
CWalks	7.3712218	4.21275888
League	-1.8663952	0.09844159
Division	-3.0683415	0.15337338
PutOuts	0.2417010	2.83947868
Assists	-0.9668011	1.45057109
Errors	1.9540215	1.21033255
NewLeague	2.0224707	0.18676702

Figure 13: Summary of Relative Influence for the predictors of the bagged regression tree

Figures 12 and 13 shows that our error rates plateaus around 200 tree for our bagged regression tree and that it's most important variables are CAtBat and CRuns. The MSE is 0.2327852, which is the lowest of models tried.