PM Spring 2024 Final Exam Question 9 Tree-based Methods

Dennis Goldenberg

5/9/2024

Note

- Change the author to your FirstName LastName
- Change the filename to TreeBaseR9.FirstLast.UNI
- Perform work and clearly write down your answers to each question
- at 4:00pm, upload both the xxxx.Rmd and its knit (word or phf)

Tree Based Question

Please refer to dataset Hitters which is in package ISLR for the question. The salary, Salary, is the response and the rest are predictors. We will use log(*Salary*) as the response when fitting the model.

The packages used for the question: tree; randomForest; gbm.

Randomly split Hitters dataset into two parts, 2/3 for training and the other 1/3 for the testing datasets. You will use the training dataset to fit the model and the testing dataset for the test error. Please set random seed to set.seed(xxxx), where xxxx is the last 4 digit of your UNI. (Hint: before create the training and testing data, you should remove any NA.)

Fit the models to the training data set and use the testing data set to obtain the mean square errors (MSEs) for the following:

- (a). Decision tree.

  i. `Fit a regression tree with the response $\log(Salary)$ to the training set.  Plot the tree. How`

  ii. What is the mean square error (MSE)?

  iii. Use cross-validation to determine the size of the pruned tree. Please plot the pruned tree. Does pruning the tree improve the test MSE?

- (b). Bagging (ntree=500) model to predict log(*Salary*). What is the test MSE? What are first two important variables? (hint: randomForest(log(Salary) ∼.,data=hitters,subset =train, ntree=500 , mtry=numvar, importance =TRUE)), where numvar = the number of predictors.)

- (c). Build random forests models to predict log(*Salary*).

  i. Build a random forest (with ntree=500, use the default value for mtry by omitting mtry) model to predict log(*Salary*) and report the test MSE. Does the MSE improve over bagging? What are the first two important variables?

  ii. Determine the optimal the number of random variables, *mtry* (each split) that has the smallest MSE by performing a for loop for mtry from 2 to 15. What is the optimal mtry?

- (d). Build Boosting trees.

  i. Build three boosting models with interaction depth *interaction.depth* = 3 (4 leaves) and default shrinkage ($\lambda = 0.1$) and ntree = 100, 500, and 1000 respectively. Obtain their test MSEs. Which of the ntree has the lowest MSE?

ii. Determine the optimal *interaction.depth*, d that has the lowest MSE by a for loop limiting the range of d from 1 to 8 by using the ntree of the lowest MSE obtained in i. above and the default shrinkage parameter. What is the optimal d?
iii. Determine the optimal *shrinkage* parameter λ that has lowest MSE by a for loop limiting the range of λ from 0.01 to 1 with 0.01 increment. Use the interaction.depth value you found in (**??**).ii. and the ntree of the lowest MSE for the ntree in (**??**).i. above. What it the optimal λ?
iv. Compare the test MSEs in (d).i. - (d).iii. above.

library

Dataset for the question.

(0). Split data into training and testing datasets and remove NA

```
set.seed(1)
#Remove NA's
na <- c()
for(i in 1:length(names(Hitters))){
  na <- union(na, which(is.na(Hitters[i])))
}
data <- Hitters[setdiff(1:dim(Hitters)[1], na),]
#Modify Response
data$Salary <- log(data$Salary)
#Generate training, test data
train_sample <- sort(sample(seq_len(nrow(data)), size =floor(2 * nrow(data)/3)))
train_Data <- data[train_sample,]
test_Data <- data[-train_sample,]
```

(a). Decision tree

- (a). Decision tree.

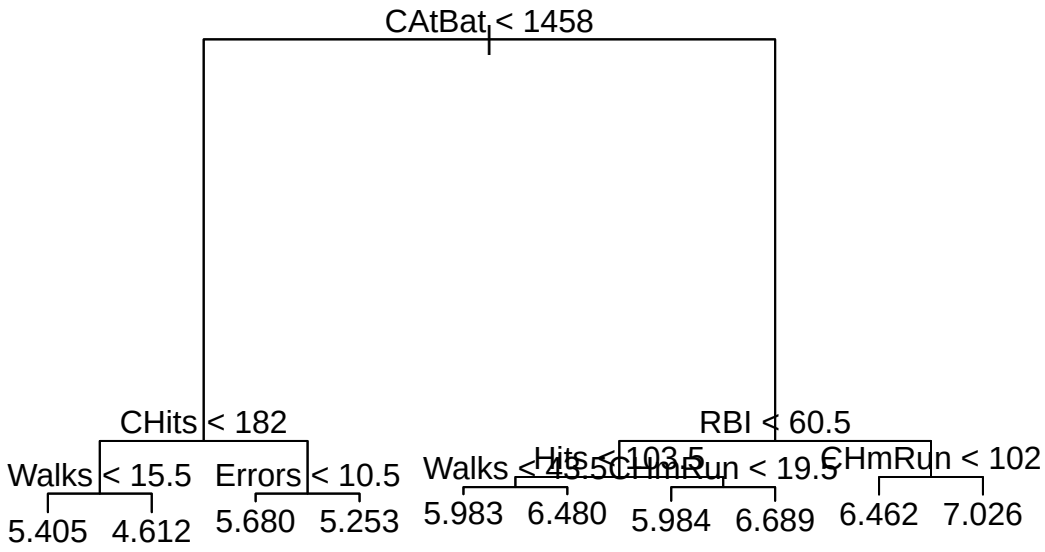
  i. Fit a regression tree with the response $\log(Salary)$ to the training set. Plot the tree. How

  ii. What is the mean square error (MSE)?
  iii. Use cross-validation to determine the size of the pruned tree. Please plot the pruned tree. Does pruning the tree improve the test MSE?

your work and answers

i.

```
rtree <- tree(Salary ~., data = train_Data)
plot(rtree)
text(rtree)
```



There are **10** terminal nodes, and the variables that are used are: CAtBat, CHits, Walks, Errors, Hits, RBI, and CHmRun. THe most important variable is **CAtBat**, or career at bats, which is split on first.

```
summary(rtree)
```

```
##
## Regression tree:
## tree(formula = Salary ~ ., data = train_Data)
## Variables actually used in tree construction:
## [1] "CAtBat" "CHits"  "Walks"  "Errors" "RBI"    "Hits"   "CHmRun"
## Number of terminal nodes:  10
## Residual mean deviance:  0.1567 = 25.86 / 165
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1.48400 -0.21110 -0.02181  0.00000  0.17890  2.25700
```

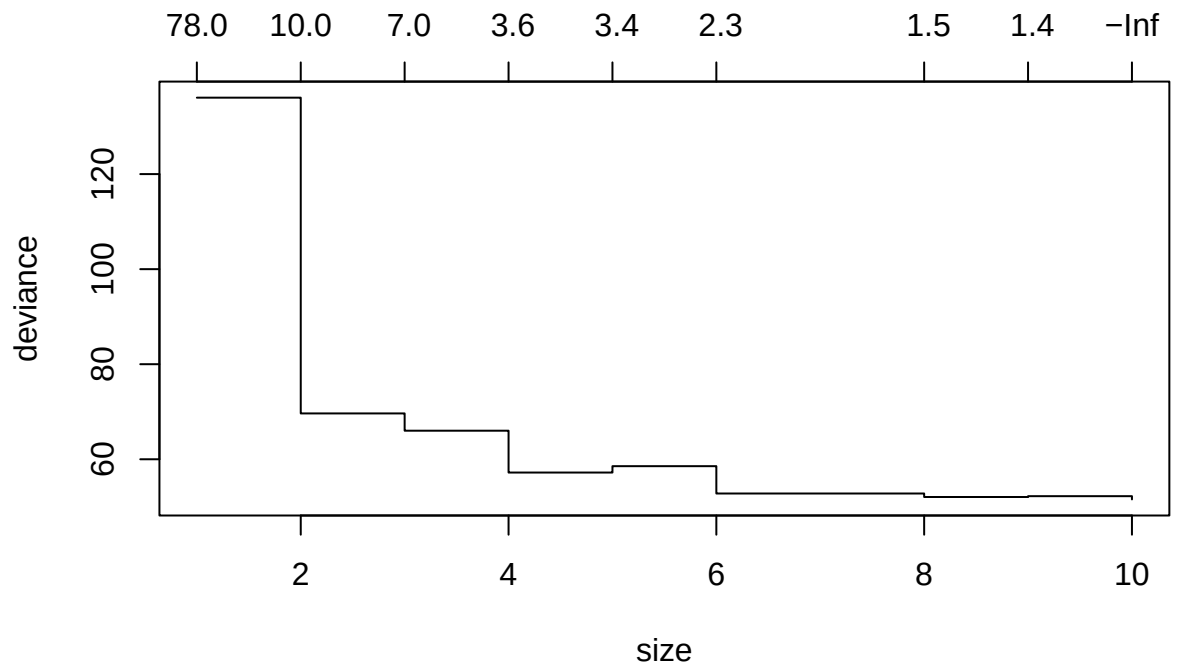
ii. What is the MSE?

```
predictions <- predict(rtree, newdata = test_Data)
MSE_tree <- sum((test_Data$Salary - predictions)^2/length(predictions))
sprintf("MSE: %.4f", MSE_tree)
```

```
## [1] "MSE: 0.3251"
```

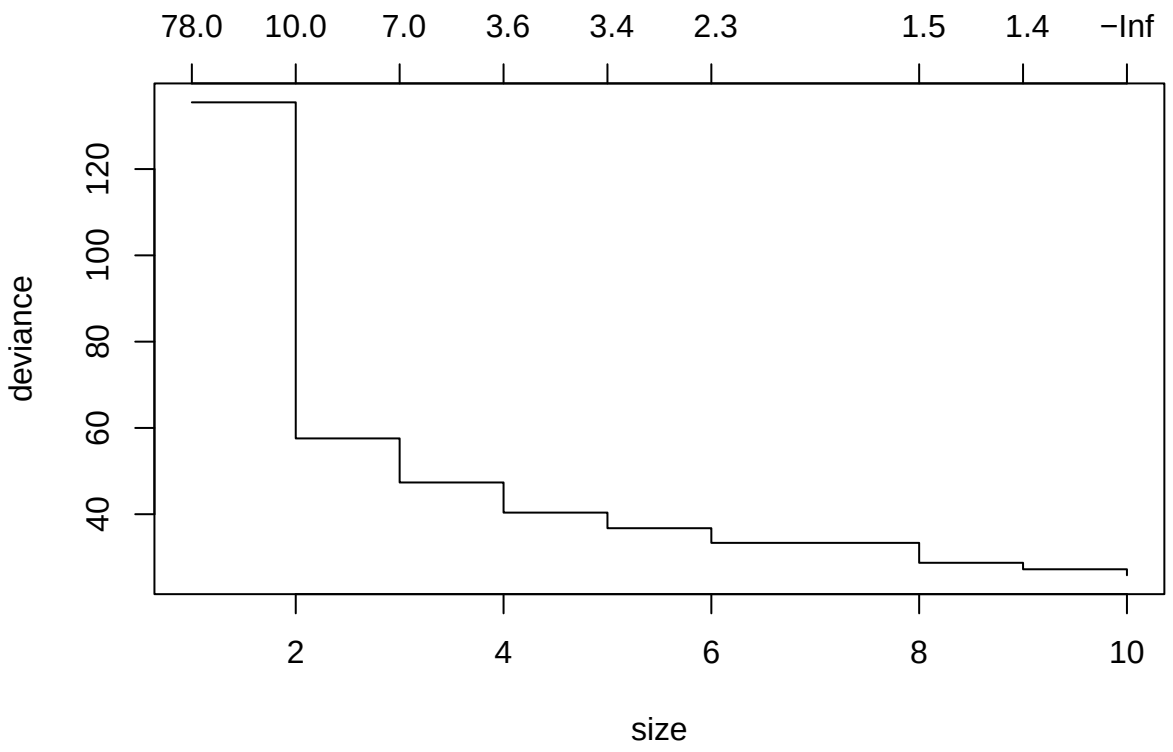
iii. Pruning the tree and plotting the optimal tree

```
cv <- cv.tree(rtree, K = 10, FUN = prune.tree)
par(oma = c(0,0,2,0))
plot(cv)
title("main = Deviance of Tree vs. Number of Leaves, corresponding alpha",
      outer = TRUE)
```



The optimal number of terminal nodes seems to be **6**. I prune the tree:

```
plot(prune.tree(rtree))
```



(b) Bagging tree

- (b). Bagging (ntree=500) model to predict log(*Salary*). What is the test MSE? What are first two important variables? (hint: randomForest(log(Salary) ∼.,data=hitters,subset =train, ntree=500 , mtry=numvar, importance =TRUE)), where numvar = the number of predictors.) I build the bagged tree:

```
bagged <- randomForest(Salary ~., data = train_Data, ntree = 500,
                       mtry = dim(data)[2] - 1, importance = TRUE)
bagged$importance
```

```
##                %IncMSE IncNodePurity
## AtBat      0.0120320668     4.6958108
## Hits       0.0062379889     4.5570955
## HmRun      0.0057305908     1.6823961
## Runs       0.0033428452     2.1654257
## RBI        0.0054214357     3.2969271
## Walks      0.0187165824     4.0387224
## Years      0.0134937298     2.1395639
## CAtBat     0.2273220233    35.3200228
## CHits      0.0996451519    14.4100298
## CHmRun     0.0167559895     4.2580336
## CRuns      0.1902382159    28.7122195
## CRBI       0.0455136677     6.7481597
## CWalks     0.0758863029    16.5549572
## League    -0.0001586032     0.1450988
## Division   0.0011907239     0.2708651
```

```
## PutOuts     0.0069405315     2.1625431
## Assists     0.0035679306     0.9741004
## Errors      0.0012683767     1.0557292
## NewLeague   0.0003410867     0.1876877
```

The most importance variables, both by the percent they increased MSE, and the percent they increased node purity, are **CAtBat**, and **CWalks**. I calculate the test MSE:

```
predict_bagged <- predict(bagged, newdata = test_Data)
MSE_bagged <- sum((predict_bagged - test_Data$Salary)^2/length(predict_bagged))
sprintf("MSE: %.4f", MSE_bagged)
```

```
## [1] "MSE: 0.2157"
```

your work and answers

(c) Random Forests

- (c). Build random forests models to predict log(*Salary*).
    i. Build a random forest (with ntree=500, use the default value for mtry by omitting mtry) model to predict log(*Salary*) and report the test MSE. Does the MSE improve over bagging? What are the first two important variables?
    ii. Determine the optimal the number of random variables, *mtry* (each split) that has the smallest MSE by performing a for loop for mtry from 2 to 15. What is the optimal mtry? ### i

```
forest <- randomForest(Salary ~., data = train_Data, ntree = 500,
                       importance = TRUE)
forest$importance
```

```
##                  %IncMSE IncNodePurity
## AtBat      0.0129663695     4.1809013
## Hits       0.0050860996     4.5611185
## HmRun      0.0069554511     1.8967829
## Runs       0.0091438259     3.1546646
## RBI        0.0079583491     3.8072019
## Walks      0.0153445795     4.5334866
## Years      0.0211262890     4.7767279
## CAtBat     0.1814137939    25.5781372
## CHits      0.1332990791    19.5242382
## CHmRun     0.0303687941     7.1611475
## CRuns      0.1408215612    21.1290259
## CRBI       0.0978218742    12.3541807
## CWalks     0.0720182535    15.2015074
## League    -0.0001347076     0.1801172
## Division   0.0008072586     0.2006061
## PutOuts    0.0070140156     2.0915729
## Assists    0.0001743032     1.0927492
## Errors     0.0017994855     0.9340638
## NewLeague  0.0004321030     0.1999655
```

This time, the two most important variables are **CAtBat** and **CRuns**, by increased node purity. I calculate the MSE:

```
predict_forest <- predict(forest, newdata = test_Data)
MSE_forest <- sum((predict_forest - test_Data$Salary)^2/length(predict_forest))
sprintf("MSE: %.4f", MSE_forest)
```

```
## [1] "MSE: 0.2056"
```

The random forest has a slight improvement over bagging, likely due to reduced variance.

your work and answers

(d) Boosting

- (d). Build Boosting trees.
    i. Build three boosting models with interaction depth *interaction.depth* = 3 (4 leaves) and default shrinkage ($\lambda = 0.1$) and ntree = 100, 500, and 1000 respectively. Obtain their test MSEs. Which of the ntree has the lowest MSE?
    ii. Determine the optimal *interaction.depth*, d that has the lowest MSE by a for loop limiting the range of d from 1 to 8 by using the ntree of the lowest MSE obtained in i. above and the default shrinkage parameter. What is the optimal d?
    iii. Determine the optimal *shrinkage* parameter λ that has lowest MSE by a for loop limiting the range of λ from 0.01 to 1 with 0.01 increment. Use the interaction.depth value you found in (**??**).ii. and the ntree of the lowest MSE for the ntree in (**??**).i. above. What it the optimal λ?
    iv. Compare the test MSEs in (d).i. - (d).iii. above.

your work and answers