Crossvalidation Labs, Regression Trees, Bagging & RandomForest Examples **Mandatory Project Check-in**

Thilanka Munasinghe

Data Analytics

ITWS-4600/ITWS-6600/MATP-4450/CSCI-4960

Group 4 - Lab 10 - November 18th, 2022

Assignment 7 available on LMS

- Assignment 7: Data Analytics (Fall 2022) (20% written)
- **Due: Tuesday, November 29th, 2022 by 11:59pm EST. (11/29/2021 by 11:59 pm ET)**
- Submission method: written by LMS Please use the following file naming for electronic submission: DataAnalytics2021Spring_A7_YOURFIRSTNAME_YOURLASTNAME.xxx, etc.
- Level: _____ (4000 or 6000)
- Late submission policy: If you are more than 5 days late it is likely that you will not have your grade for this assignment included in your final grade before they need to be submitted. First time with valid reason – no penalty, otherwise 20% of score deducted each late day.

- Note: Your assignment should be the result of your own individual work. Take care to avoid plagiarism ("copying"), and include references to all web resources, texts, and class presentations. You may discuss the project with other students, but do not take written notes during these discussions, and do not share your written assignment or presentation before the class they are presented in.
- General assignment: Predictive and Prescriptive data analytics. You should develop and validate predictive models (regression, classification, clustering – using one or more of the methods covered in class to date or one of your choosing) for ***two*** of the eight (the Wine Quality contains red wine and white wine datasets, both red and white wine dataset counts as one dataset) datasets below and apply them for decision purposes. Use the section numbering below for your written submission for this assignment. Include references – websites, papers, packages, data refs...

Revisiting Regression

- Now we are revisiting regression that we learned during the Group 1 (beginning of the semester) and now we will use a tree-based regression technique during this example.
- **RandomForest regressor can be used conduct the regression.**

Fitting a Regression Trees

```
library(MASS)
library(tree)
set.seed(1)
head(Boston)
train = sample(1:nrow(Boston), nrow(Boston)/2)
tree.boston = tree(medv ~., Boston, subset = train)
summary(tree.boston)
# Note that the output summary() indicates that only three of the variables have been
# used to constructing the tree. In the context of a regression tree,
# the deviance is simply the sum of squared errors for the tree.
```


```r
# Fitting a Regression Trees

library(MASS)
library(tree)
set.seed(1)
head(Boston)
train = sample(1:nrow(Boston), nrow(Boston)/2)
tree.boston = tree(medv ~., Boston, subset = train)
summary(tree.boston)
# Note that the output summary() indicates that only three of the variables have been
# used to constructing the tree. In the context of a regression tree,
# the deviance is simply the sum of squared errors for the tree.

# Regression Tree
tree(formula = medv ~. , data = Boston, subset = train)
```


Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

```
# Regression Tree
tree(formula = medv ~. , data = Boston, subset = train)

# We now plot the tree
plot(tree.boston)
text(tree.boston, pretty = 0)
# The variable "lstat" measure the percentage of the individuals with lower socioeconomics status.
# The tree indicates that the lower values of lstat corresponds to more expensive house.

# Now we use the cv.tree() function to see whether pruning the tree will
# improve performance.
cv.boston=cv.tree(tree.boston)
plot(cv.boston$size ,cv.boston$dev ,typ ='b')
```


```
# Regression Tree
tree(formula = medv ~. , data = Boston, subset = train)

# We now plot the tree
plot(tree.boston)
text(tree.boston, pretty = 0)
# The variable "lstat" measure the percentage of the individuals with lower socioeconimics status.
# The tree indicates that the lower values of lstat corresponds to more expensive houese.

# Now we use the cv.tree() function to see whether pruning the tree will
# improve performance.
cv.boston=cv.tree(tree.boston)
plot(cv.boston$size ,cv.boston$dev ,typ ='b')
```


Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

#In this case, the most complex tree is selected by cross-validation.
However, if we wish to prune the tree, we could do so as follows,
#using the prune.tree() function

prune.boston=prune.tree(tree.boston ,best=5)
plot(prune.boston)
text(prune.boston ,pretty=0)

```
#In this case, the most complex tree is selected by cross-validation.
# How- ever, if we wish to prune the tree, we could do so as follows,
#using the prune.tree() function

prune.boston=prune.tree(tree.boston ,best=5)
plot(prune.boston)
text(prune.boston ,pretty=0)
```


Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

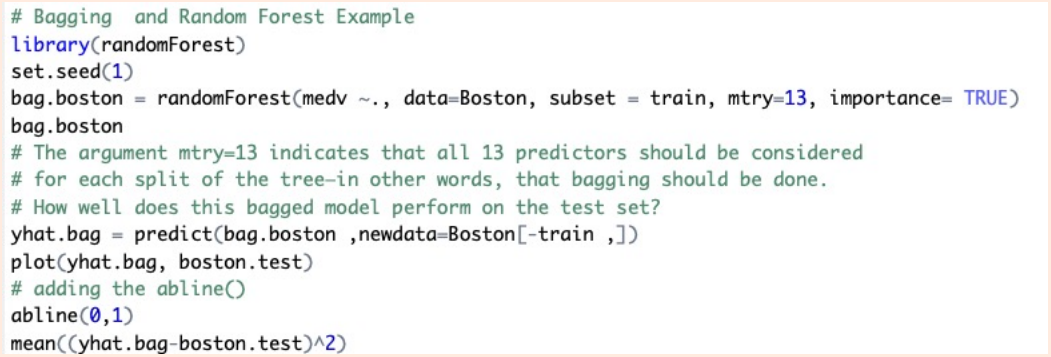
# In keeping with the cross-validation results,
# we use the unpruned tree to make predictions on the test set.
yhat=predict(tree.boston ,newdata=Boston[-train ,])
boston.test=Boston[-train ,"medv"]
plot(yhat,boston.test)
# adding the abline()
abline(0,1)
mean((yhat-boston.test)^2)
# In other words, the test set MSE associated with the regression tree is 25.05.
# The square root of the MSE is therefore around 5.005,
# indicating that this model leads to test predictions that
# are within around $5, 005 of the true median home value for the suburb.


```
# In keeping with the cross-validation results,
# we use the unpruned tree to make predictions on the test set.
yhat=predict(tree.boston ,newdata=Boston[-train ,])
boston.test=Boston[-train ,"medv"]
plot(yhat,boston.test)
# adding the abline()
abline(0,1)
mean((yhat-boston.test)^2)
# In other words, the test set MSE associated with the regression tree is 25.05.
# The square root of the MSE is therefore around 5.005,
# indicating that this model leads to test predictions that
# are within around $5, 005 of the true median home value for the suburb.
```


Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

Bagging & RandomForest Example

```
# Bagging  and Random Forest Example
library(randomForest)
set.seed(1)
bag.boston = randomForest(medv ~., data=Boston, subset = train, mtry=13, importance= TRUE)
bag.boston
# The argument mtry=13 indicates that all 13 predictors should be considered
# for each split of the tree—in other words, that bagging should be done.
# How well does this bagged model perform on the test set?
yhat.bag = predict(bag.boston ,newdata=Boston[-train ,])
plot(yhat.bag, boston.test)
# adding the abline()
abline(0,1)
mean((yhat.bag-boston.test)^2)
```



Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

The test set MSE associated with the bagged regression tree is 13.16,
#almost half that obtained using an optimally-pruned single tree.
#We could change the number of trees grown by randomForest() using the ntree argument:
bad.boston =
bag.boston=randomForest(medv~.,data=Boston,subset=train, mtry=13,ntree=25)
yhat.bag = predict(bag.boston ,newdata=Boston[-train ,])
mean((yhat.bag-boston.test)^2)

```
# The test set MSE associated with the bagged regression tree is 13.16,
#almost half that obtained using an optimally-pruned single tree.
#We could change the number of trees grown by randomForest() using the ntree argument:
bad.boston =
bag.boston=randomForest(medv~.,data=Boston,subset=train, mtry=13,ntree=25)
yhat.bag = predict(bag.boston ,newdata=Boston[-train ,])
mean((yhat.bag-boston.test)^2)
```


Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

```
set.seed(1)
rf.boston=randomForest(medv~.,data=Boston,subset=train,
                       mtry=6,importance =TRUE)
yhat.rf = predict(rf.boston ,newdata=Boston[-train ,])
mean((yhat.rf-boston.test)^2)
# The test set MSE is 11.31;
# this indicates that random forests yielded an improvement over bagging in this case.
# Using the importance() function, we can view the importance of each variable.
```


Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

importance (rf.boston)
Two measures of variable importance are reported.
#The former is based upon the mean decrease of accuracy in predictions on
the out of bag samples when a given variable is excluded from the model.
#The latter is a measure of the total decrease in node impurity that results
from splits over that variable, averaged over all trees.

In the case of regression trees, the node impurity is measured by the training RSS,
and for classification trees by the deviance.
Plots of these importance measures can be produced using the varImpPlot() function.
varImpPlot (rf.boston)

```
importance (rf.boston)
# Two measures of variable importance are reported.
#The former is based upon the mean decrease of accuracy in predictions on
# the out of bag samples when a given variable is excluded from the model.
#The latter is a measure of the total decrease in node impurity that results
# from splits over that variable, averaged over all trees.

# In the case of regression trees, the node impurity is measured by the training RSS,
# and for classification trees by the deviance.
# Plots of these importance measures can be produced using the varImpPlot() function.
varImpPlot (rf.boston)
```


Reference/Resource: Introduction to Statistical Learning with R; Chapter 8

Group 4 Labs: Crossvalidation

- Work on the Group 4 Cross Validation Labs:
- Lab CV_1, Lab CV_2
- Lab CV_3, Lab CV_4, Lab CV_5,
- Lab CV_6
- ….
- Lab CV_18
- **NOTE: you are allowed to work in small groups and discuss during this lab.**

script fragments in R available on Group 4:
https://rpi.box.com/s/b0jzxaeaqmhgx67y5jei9x1nv3d6wh11

Reminder: Work through these code snippets if you have not yet finished them…

- lab1_svm1.R
- lab1_svm2.R
- lab1_svm3.R
- lab1_svm4.R
- lab1_svm5.R
- lab1_svm6.R
- lab1_svm7.R
- lab1_svm8.R
- lab1_svm9.R
- lab1_svm10.R
- lab1_svm12.R
- lab1_svm13.R
- lab1_svm_rpart1.R

NOTE: you are allowed to work in small groups and discuss during these labs.

These code snippets are available in the course repository:

script fragments in R available on Group 3 and 4:
https://rpi.box.com/s/kk6f5gp9oq56wicgdvilgw5cokibh7uu

Trees for the Titanic

Reminder to complete the following if you have not done yet…
After you complete this task, push your code to GitHub,

data(Titanic)

rpart, ctree, hclust, RandomForest for:
Survived ~ .

Today…

Mandatory Project Check-in during the Lab:

One-on-One with the Instructor on Project Progress Updates during the Labs. You Must check-in with the Instructor. Instructor will give the feedback on your project progress.

- You need to meet with instructor and show your current progress and the development of your term project (Assignment6).