Math 154 Tree-based Methods Lab

Roja, Franklin, Alejandra

November 13, 2016

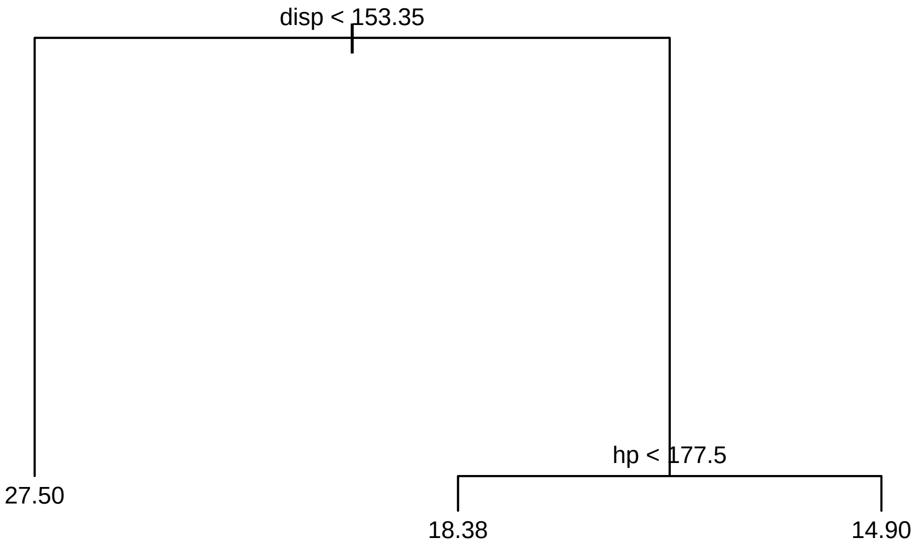
In addition to using the kaggle data, load the data set mtcars into R via the command data(mtcars). Try to predict the fuel efficiency (mpg) via a regression tree, and try to predict the transmission type (automatic or manual) via a classification tree. You may want to compare the regression tree to a linear model. plot(mtcars) will give you all the pairwise scatter plots. Notice that most of the relationships with MPG are non-linear.

Additionally, R has a package called randomForest. The most useful function, which implements the algorithm discussed in class, is of the same name. Compare the classification rates for a random forest to that of a simple tree.

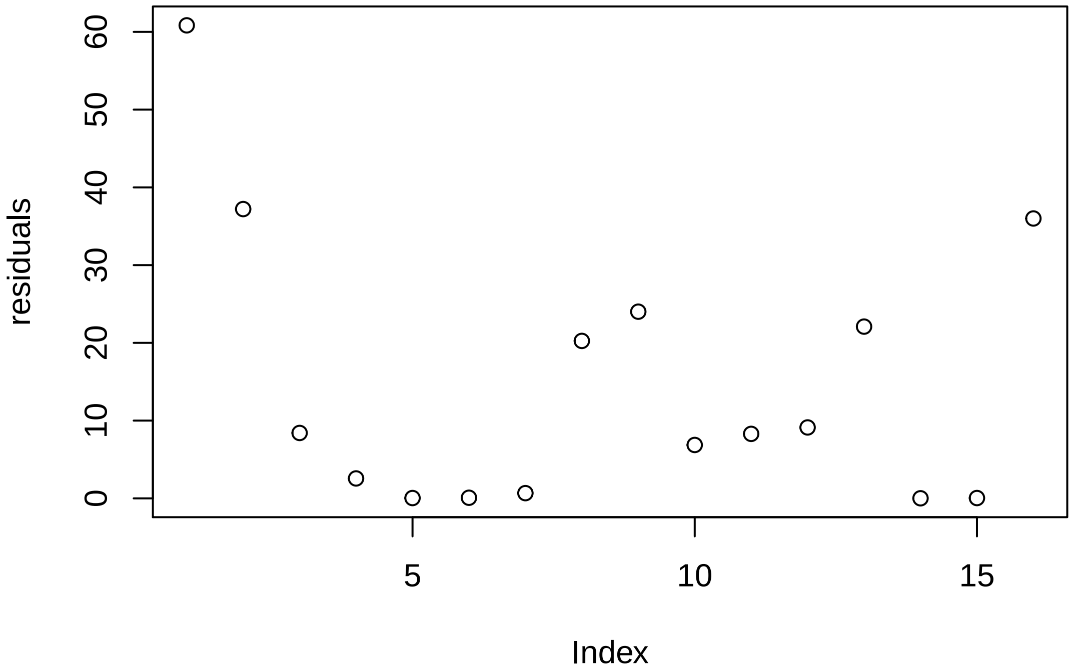
```r
# install.packages("tree")
# install.packages("randomForest")
# install.packages('reprtree')
library(tree)
library(randomForest)

## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.

library(datasets)
# library(reprtree)

data(mtcars)
```

In order to predict the fuel efficiency we implemented a regression tree and a linear regression model to later compare results. We divided the data into a training and test set. Since the sample size is only thirty-two, the two sets only had sixteen observations. One method we used to compare the performance of both models was residual sum of squares. We consistently found that predictions made with a linear regression model had a lower residual sum of squares than those made with a regression tree. It may be due to there only being three nodes and therefore, three averages which new observations can take on.

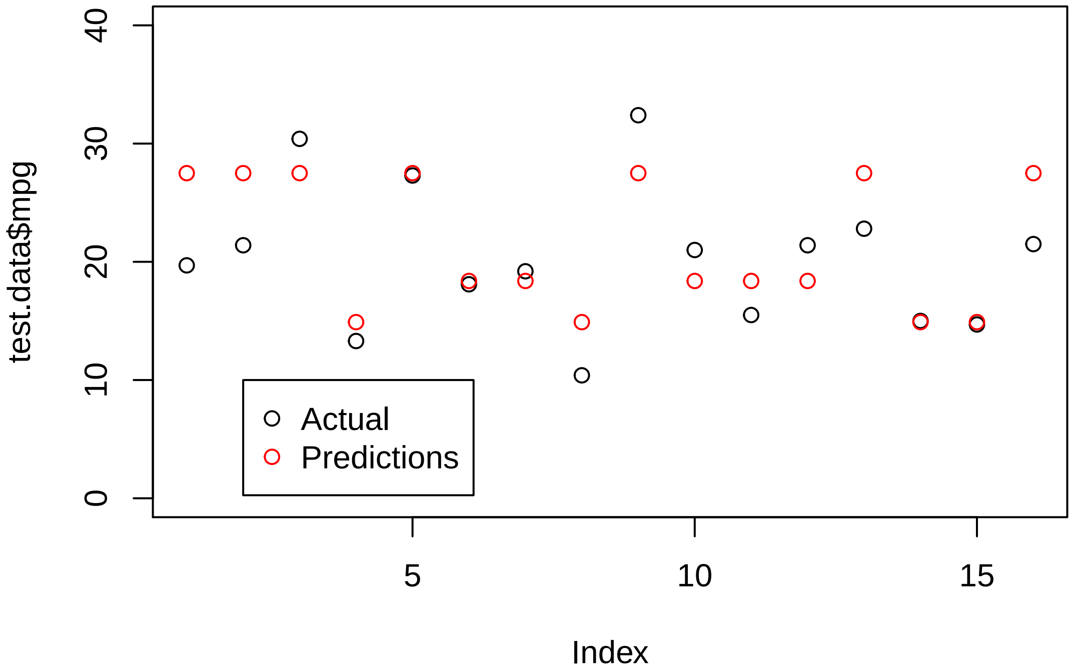
```r
# Regression Tree
test.index = sample(c(1:nrow(mtcars)),nrow(mtcars)/2 )
training.index= c(1:nrow(mtcars))[-test.index]

test.data = mtcars[test.index,]
training.data = mtcars[training.index,]

cars.regression <- tree(mpg ~ cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb, data=training.data)
plot(cars.regression )
text(cars.regression , cex=.75)
```



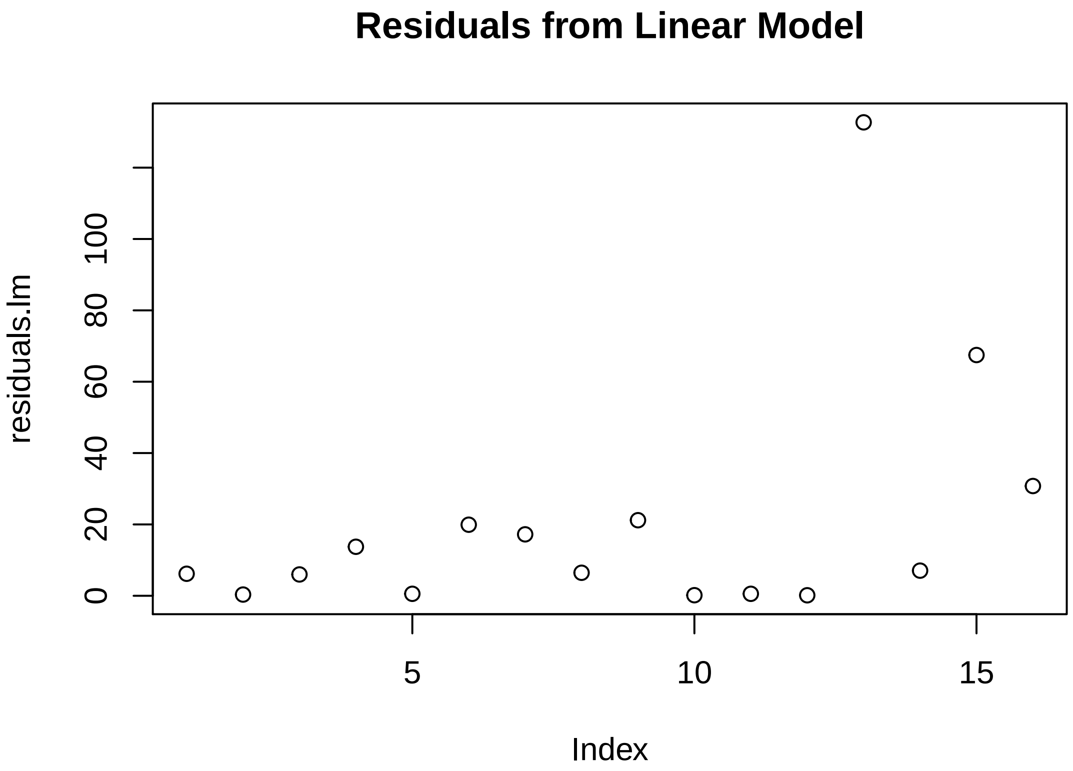
```
my.prediction <- predict(cars.regression, test.data)

# find RSS
residuals = (test.data$mpg - my.prediction)^2
sum(residuals^2)

## [1] 8133.442

# plot residuals
plot(residuals)
```



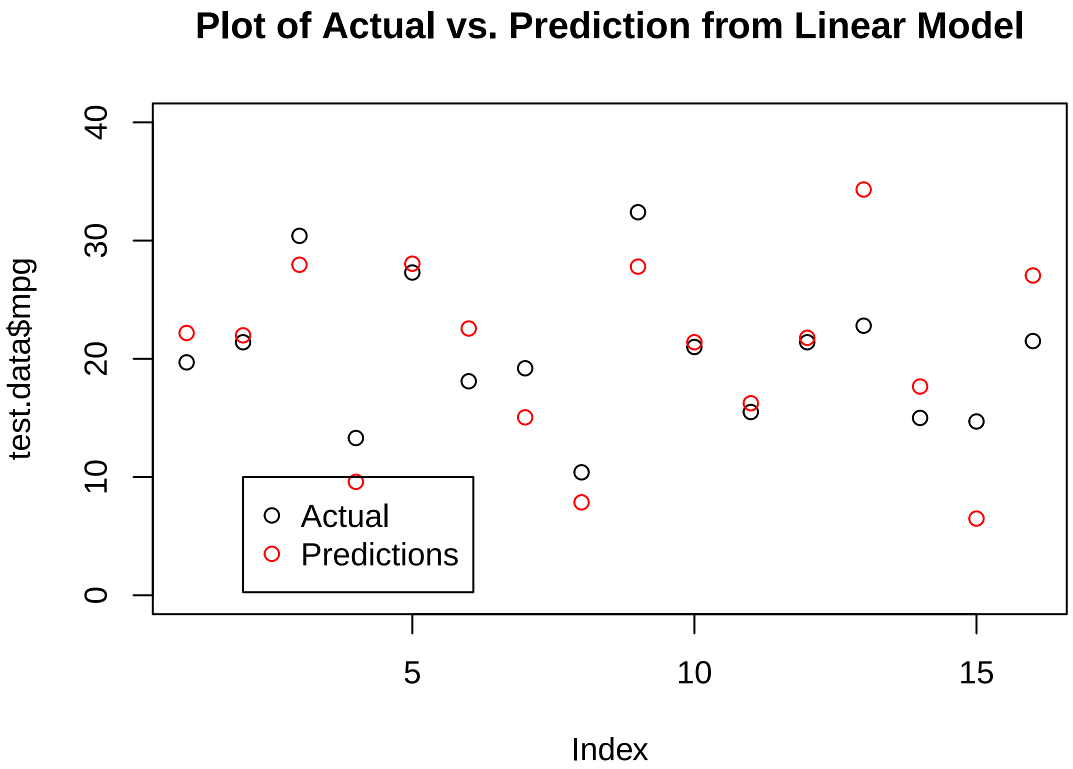
```
# plot of actual and predictions
plot(test.data$mpg, ylim = c(0,40))
points(my.prediction, col = 'red')
legend(2, 10, legend=c("Actual", "Predictions"),
       col=c("black", "red"),pch = 1)
```



```
#####
# Comparing to a Linear model
#plot(mtcars)

cars.lm <- lm ( mpg ~ cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb, data= training.data)
prediction.lm <- predict(cars.lm, test.data)
# find RSS
residuals.lm = (test.data$mpg - prediction.lm)^2
sum(residuals.lm^2)

## [1] 24604.64

# plot residuals
plot(residuals.lm, main = "Residuals from Linear Model")
```



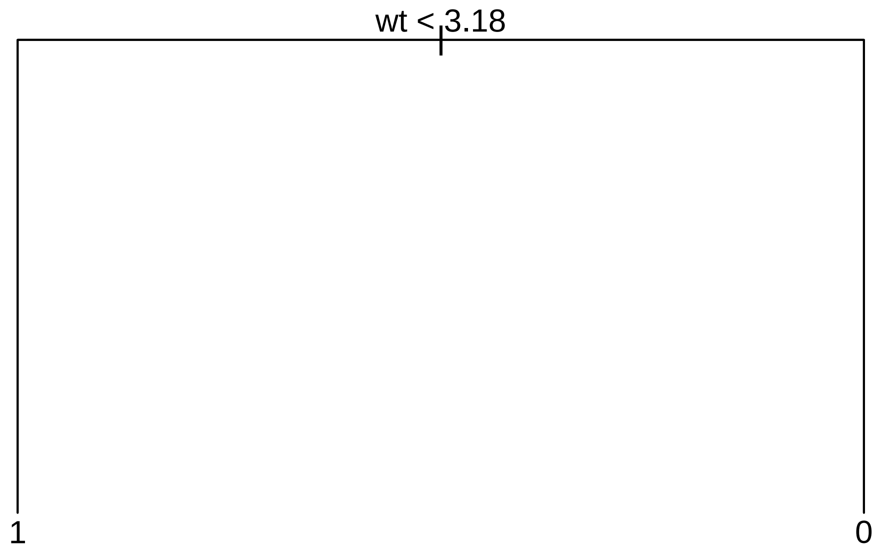
```
# plot of actual and predictions
plot(test.data$mpg, main= "Plot of Actual vs. Prediction from Linear Model", ylim = c(0,40))
points(prediction.lm, col = 'red')
legend(2, 10, legend=c("Actual", "Predictions"),
       col=c("black", "red"),pch = 1)
```



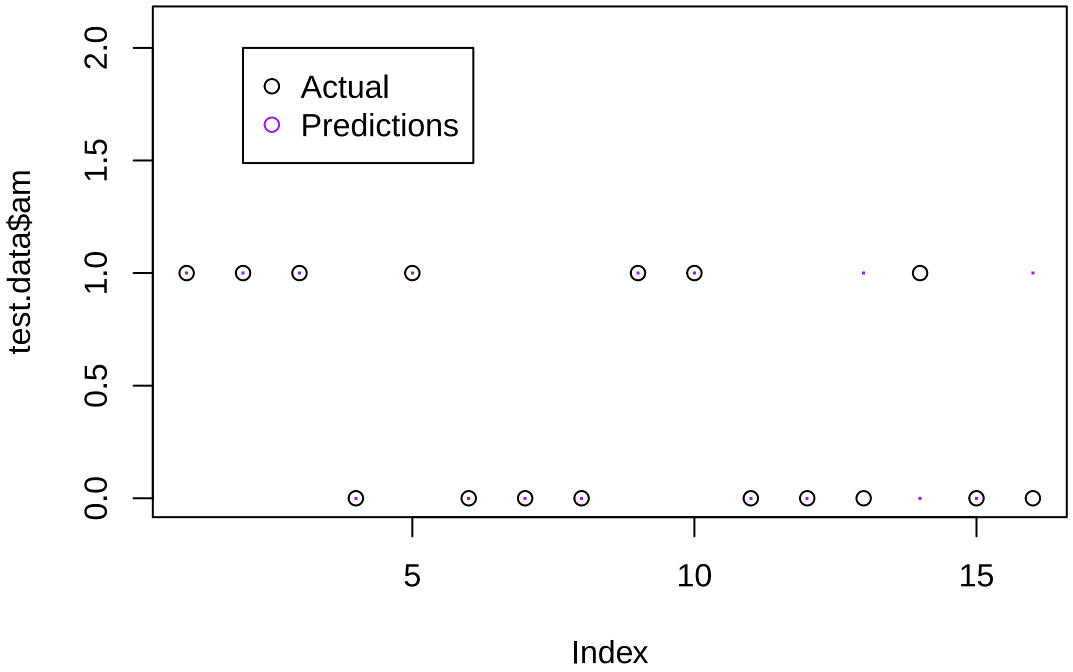
```
# to do:
# reduce the number f variables? stepwise regression?
# consider nonlinear relationships
```

We used Classification and Random Forests to predict the transmission type of a car. The classification tree had three terminal nodes and used *weight* and *mpg* in the tree. Since the response variable was binary we calculated the proportion of times each method predicted accurately. The accuracy of the predictions from both methods were about the same.

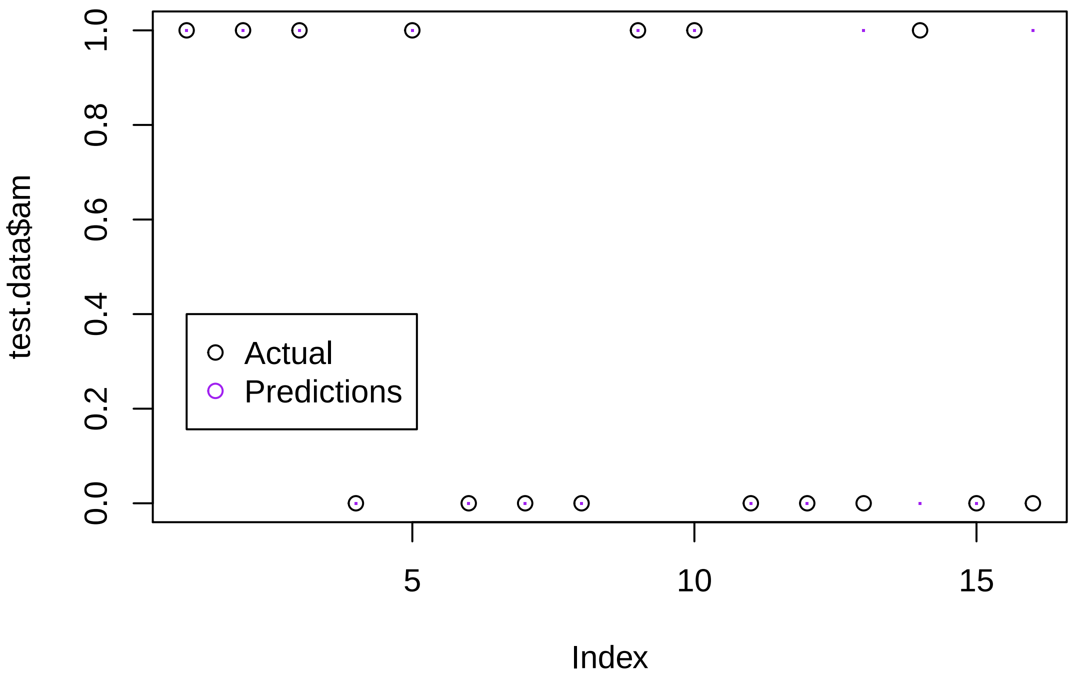
```
# Classification Tree
# Use a classification tree to predict transmission type of car
# Transmission (0 = automatic, 1 = manual)

cars.class<-tree(am ~ mpg+cyl+disp+hp+drat+wt+qsec+vs+gear+carb, data=training.data)
summary(cars.class)

##
## Regression tree:
## tree(formula = am ~ mpg + cyl + disp + hp + drat + wt + qsec +
##     vs + gear + carb, data = training.data)
## Variables actually used in tree construction:
## [1] "wt"
## Number of terminal nodes:  2
## Residual mean deviance:  0 = 0 / 14
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0

plot(cars.class)
text(cars.class)
```



```
my.prediction.class <- predict(cars.class, test.data)
plot(test.data$am, ylim = c(0,2.1))
points(my.prediction.class,col = 'purple', pch = ".")
legend(2, 2, legend=c("Actual", "Predictions"),
       col=c("black", "purple"),pch = 1)
```



```
# how many incorrect predictions
incorrect.predict = sum(abs(my.prediction.class - test.data$am) > .5)
incorrect.predict

## [1] 3

####
# Using a Random Forest

cars.Forest <- randomForest(as.factor(am) ~ mpg+cyl+disp+hp+drat+wt+qsec+vs+gear+carb, data=training.data ,
                            ntree=128)
prediction.Forest<- predict(cars.Forest, test.data)

plot(test.data$am)
points(as.numeric(prediction.Forest)-1,col = 'purple', pch = ".")
legend(1, 0.4, legend=c("Actual", "Predictions"),
       col=c("black", "purple"),pch = 1)
```



```
incorrect.predict.Forest = sum(abs((as.numeric(prediction.Forest) - 1) - test.data$am) > .5)
incorrect.predict.Forest

## [1] 3
```