Linear Regression or Regression Tree?

Keywords: regression tree, linear regression, MSE/RMSE.

Background

Simple linear regression and regression tree both can be used in supervised learning, then a natural question to ask is: which one to choose? Even though one could apply both model and select the one performs better, are there any general preferences in choosing models? Especially when the response variable is interval. I decide to test which one works better by doing a comparison. In order to make things simple I will just compare simple tree regression to simple linear regression instead of using more complicated random forest for now.

The dataset I use is from famous UCI automobile dataset. You can find the data at https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data

The purpose is to predict car price using a series of interval and nominal variables such as car length, width, height, curb-weight, compression-ratio, horsepower, mpg, fuel-type, body-style, drive-wheels just to name a few.

In order to compare which model is better, I have to select a criterion. For now I am using MSE (mean-square error). The smaller the MSE, the better the model.

To simplify the process, I only use rows that do not contain missing values.

I will walk through the key codes and ignore the trivial ones but include a copy of my code along with this report to GitHub.

The tree model

```
library(data.table)

library(tree)

mydata <- fread(paste("https://archive.ics.uci.edu","/ml/machine-learning-databases/autos/imports-85.data",sep=""))
```

These lines above use tree and data.table libraries. mydata is the original automobile data set on the UCI website.

```
mydata2[,c(3:9,15,16)] <- lapply(mydata2[,c(3:9,15,16)],factor)

mydata2[,c(10:14,17,19:26)] <- lapply(mydata2[,c(10:14,17,19:26)],as.numeric)
```

Then convert character variables into factors using lapply function as r handles nominal variables as factors. Using the same idea to convert character variables into numeric ones.

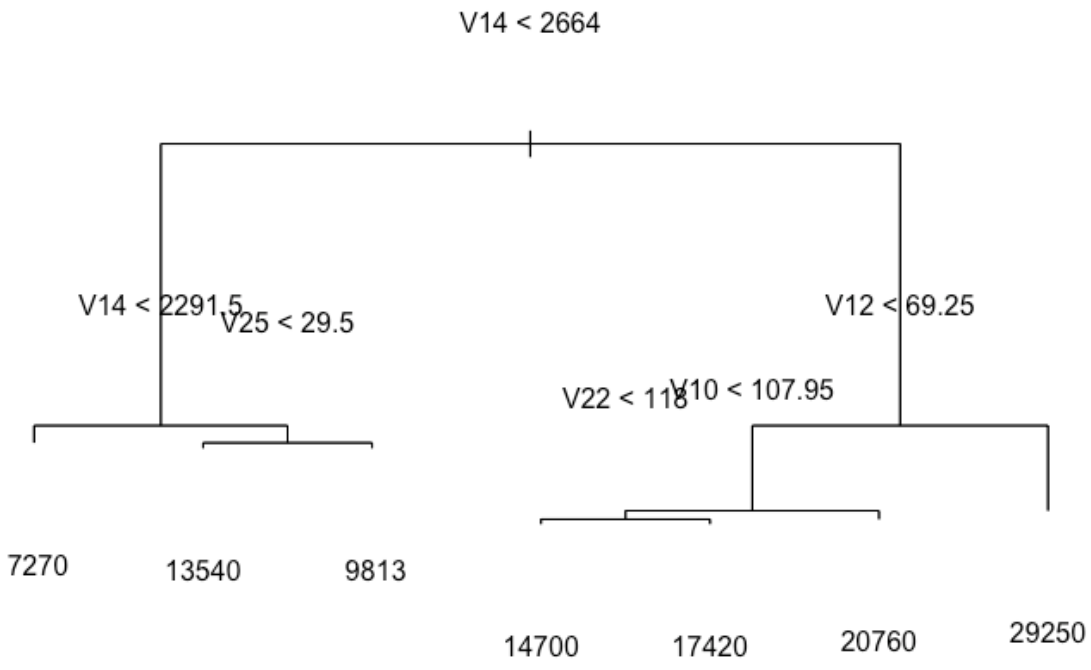
```
train <- sample(1:nrow(mydata2),0.7*nrow(mydata2))

train_data <- mydata2[train]

test_data <- mydata2[-train]
```

The above codes split data set into training (70%) and holdout (30%).

```
tree_model <- tree(V26~V4+V5+V6+V7+V8+V10+V11+V12+V13+V14+V15+V16+V17+V19+V20+V21+V22+V23+V24+V25,train_data)

plot(tree_model)

text(tree_model,pretty=0)
```

A tree model is built and tree chart plotted below. Note I did not use V3 (car brand), V9 (engine location: front or rear. V9 all value equals to front so no predicting power), or V18.

We can see from the tree chart below that without pruning, car weight (V14) has the most differentiating ability, followed by mpg-highway (V25), car width (V12), wheel-base (V10), and horsepower (V22). Heavy car is more expensive than light car. Not sure why lower highway mpg car is more expensive than higher mpg. Wide car is more expensive, and cars have high horsepower is more expensive. The longer the wheelbase the pricy the car will be. In sum, bigger is better! Remember the data set was created in 1985 so it only reflects the 80s preference.



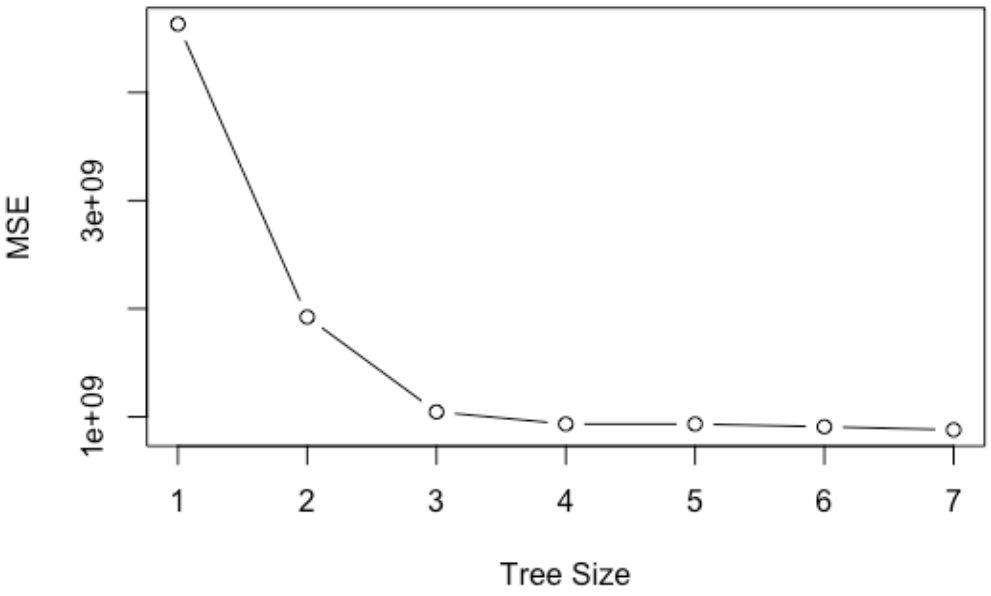

```
cv_tree <- cv.tree(tree_model)

names(cv_tree)

plot(cv_tree$size,cv_tree$dev,type="b", xlab="Tree Size",ylab="MSE")
```

The above codes find the "optimal" tree leaf size by comparing MSEs of different number of tree leaves. From the chart below, it seems keep the number of leaves to 7 yields the minimum MSE, pruning will increase MSE. Therefore, no pruning is necessary. The names(cv_tree) function shows the elements in the cv_tree object. There are four elements in cv_tree object: tree size, deviation from the actual value, k, and method.

```
> names(cv_tree)
```

[1] "size" "dev" "k" "method"

tree_pred <- predict(tree_model,test_data)




```
mean((tree_pred-test_data$V26)^2) # result is 6,662,722
```

Next step I make prediction using the hold-out data set and find the MSE and RMSE. MSE=6,662,722, RMSE=2,581.

Before performing linear regression and comparing its MSE/RMSE with regression tree's MSE/RMSE, notice there are only 7 "leaves" in the tree model, meaning we only have 7 car prices if we use the tree model, which seems not very precise considering many levels of price we had in the original data set.

Simple regression model

Now let us use regular simple regression to predict the car price and find the RMSE/MSE. The code this time is much simpler.

```
lr_model <-
lm(V26~V4+V5+V6+V7+V8+V10+V11+V12+V13+V14+V15+V16+V17+V19+V
20,train_data)

lr_predict <- predict(lr_model,test_data)

mean((lr_predict-test_data$V26)^2) #result is 3,501,887
```

The same variables are used in the model as in the tree model. Then predictions are made using hold-out data set and RMSE/MSE is calculated.

```
summary(lr_model)

library(QuantPsyc)

lm.beta(lr_model)
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1954 on 85 degrees of freedom
Multiple R-squared:  0.9284,    Adjusted R-squared:  0.9074
F-statistic: 44.11 on 25 and 85 DF,  p-value: < 2.2e-16
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -13832.507  17384.758  -0.796  0.42844
V4gas         1083.495    876.591   1.236  0.21985
V5turbo       1159.593    752.565   1.541  0.12707
V6two          399.544    655.065   0.610  0.54354
V7hardtop    -2906.617   2342.434  -1.241  0.21807
V7hatchback  -1612.035   2204.632  -0.731  0.46667
V7sedan       -393.972   2262.892  -0.174  0.86220
V7wagon      -1860.577   2297.829  -0.810  0.42037
V8fwd         -275.602   2463.389  -0.112  0.91118
V8rwd          387.440   2616.689   0.148  0.88264
V10             93.425    105.476   0.886  0.37826
V11           -118.333     59.354  -1.994  0.04939 *
V12            511.988    291.546   1.756  0.08267 .
V13            -73.868    164.053  -0.450  0.65366
V14              8.704      2.610   3.335  0.00126 **
V15l         -3364.057   1624.085  -2.071  0.04136 *
V15ohc        1021.645   1053.521   0.970  0.33493
V15ohcf      -3761.758   2036.348  -1.847  0.06818 .
V15ohcv      -3404.295   1528.935  -2.227  0.02862 *
V16five      -5969.421   3801.785  -1.570  0.12009
V16four     -11001.587   4037.032  -2.725  0.00780 **
V16six       -8698.965   3280.294  -2.652  0.00955 **
V16three     -3551.231   5103.288  -0.696  0.48841
V17             16.312     33.347   0.489  0.62599
V19           1634.659   1830.901   0.893  0.37448
V20          -3599.093   1188.926  -3.027  0.00327 **
---
```

```
> lm.beta(lr_model)
        V4gas       V5turbo         V6two     V7hardtop   V7hatchback       V7sedan       V7wagon
 5.264115e-02  7.353623e-02  3.104192e-02 -3.490011e-01 -1.243758e-01 -3.414989e-01 -3.473785e+00
        V8fwd         V8rwd           V10           V11           V12           V13           V14
-8.794702e-02  1.356033e-01  7.504130e+00 -1.438350e-02  3.768018e-02 -3.821645e-01  3.670648e-04
         V15l        V15ohc       V15ohcf       V15ohcv       V16five       V16four        V16six
-1.493250e-01  4.963620e-02 -2.385539e-01 -2.644914e-01 -7.167559e-01 -8.488221e-01 -7.540348e+00
     V16three           V17           V19           V20
-6.630318e+00  5.205299e-03  5.721278e-01 -2.890892e+02
```

From the model diagnosis, the model is significant at 0.05 level (p <2.2e -16). Adjusted R-square is 0.91 means 91% of the variance can be explained

by the model. MSE= 3,501,887, RMSE=1,871, both numbers are much smaller than those from tree regression as showed before.

In order to see which variables are more important, we have to see their standardized coefficients. We use library(QuantPsyc) in r. We can see that wheel-base (V10=7.50) has biggest positive influence, followed by bore (V19=0.57). Six cylinder has biggest negative influence (V16six=-7.5). As a result, the regression tree and linear regression do not necessarily pick the same variables to be most influential. However, most of the coefficients are insignificant in the linear regression. Notice that there is some overlap between significant variables in the linear regression and variables selected by regression tree.

Conclusion

It looks like from this simple example that linear regression, when used properly, can yield better prediction than using simple tree regression when the response variable is interval variable (car price, life expectancy, customer scores). Linear regression gives more precise prediction values and has lower RMSE/MSE compared with regression tree model. However, a best practice may be to test both models using cross validation and make a decision. Further study is need to investigate whether random forest will perform better than simple linear regression.