Linear Regression or Regression Tree?

Background

Simple linear regression and regression tree both can be used in supervised learning, 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 preferences in choosing models 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 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.

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 of converting 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%).

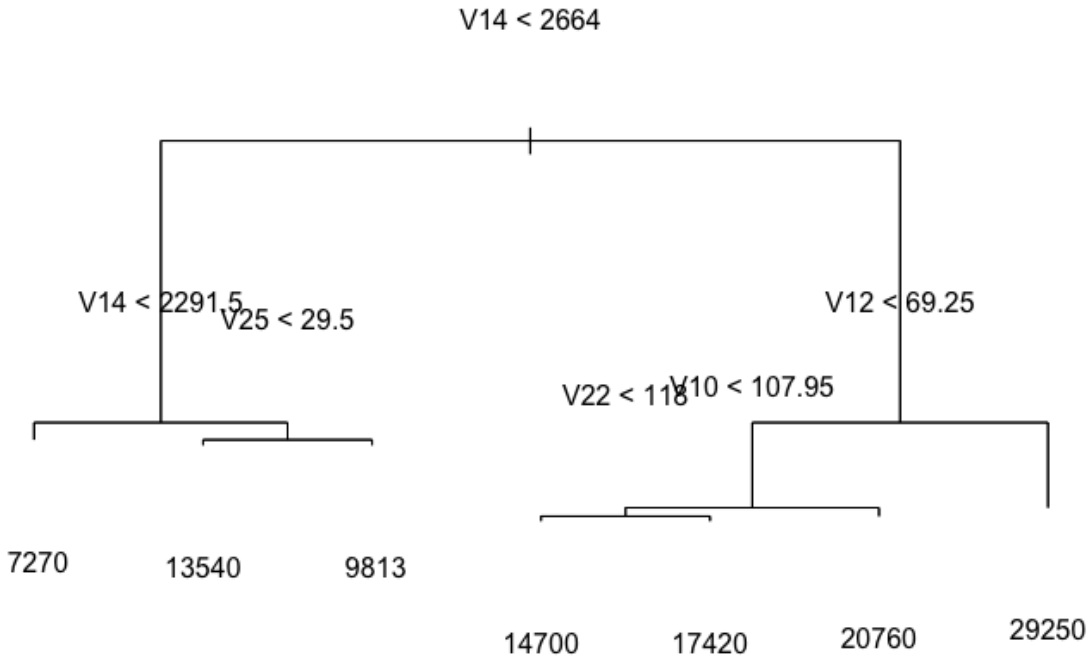
```
tree_model <- tree(V26~V4+V5+V6+V7+V8+V10+V11+V12+V13+V14+V15+V16+V17+V19+V20+V21+V22+V23+V24+V25,train_data)

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).




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

names(cv_tree)

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

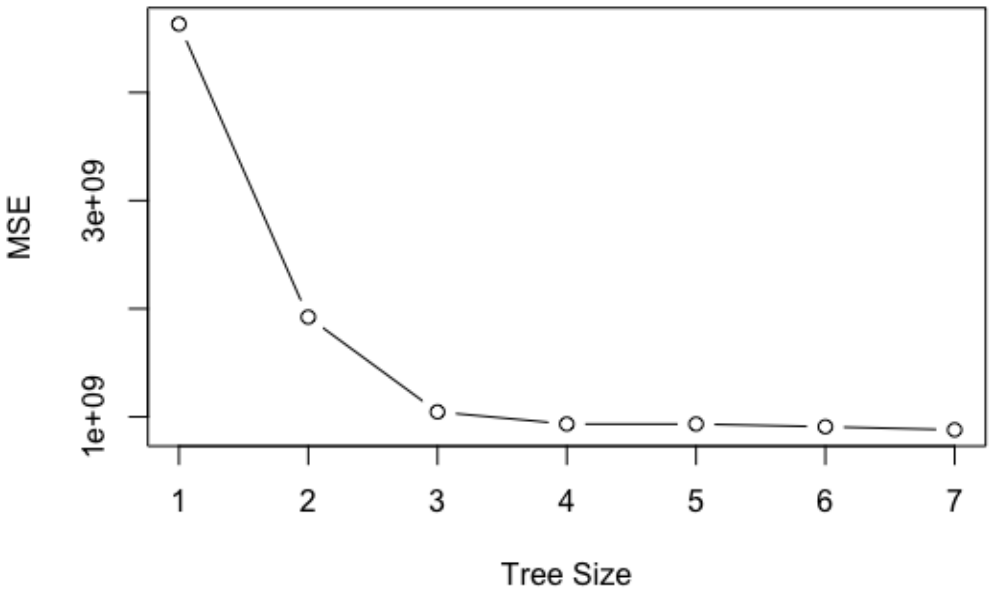
The above codes find the "optimal" tree leaf size by comparing MSEs of different number of tree leaves. From the chart below, it seems keep the number of leaves to 7 has the minimum MSE, pruning will increase MSE value. Therefore, no pruning is necessary. The names(cv_tree) function

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 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 using 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.

```
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
```

From at 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.

Conclusion

It looks like from this simple example that linear regression, when used properly, could give 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, further study is need to investigate whether random forest will perform better than simple linear regression.