Predict Cars Prices Regression Tree

Lucky

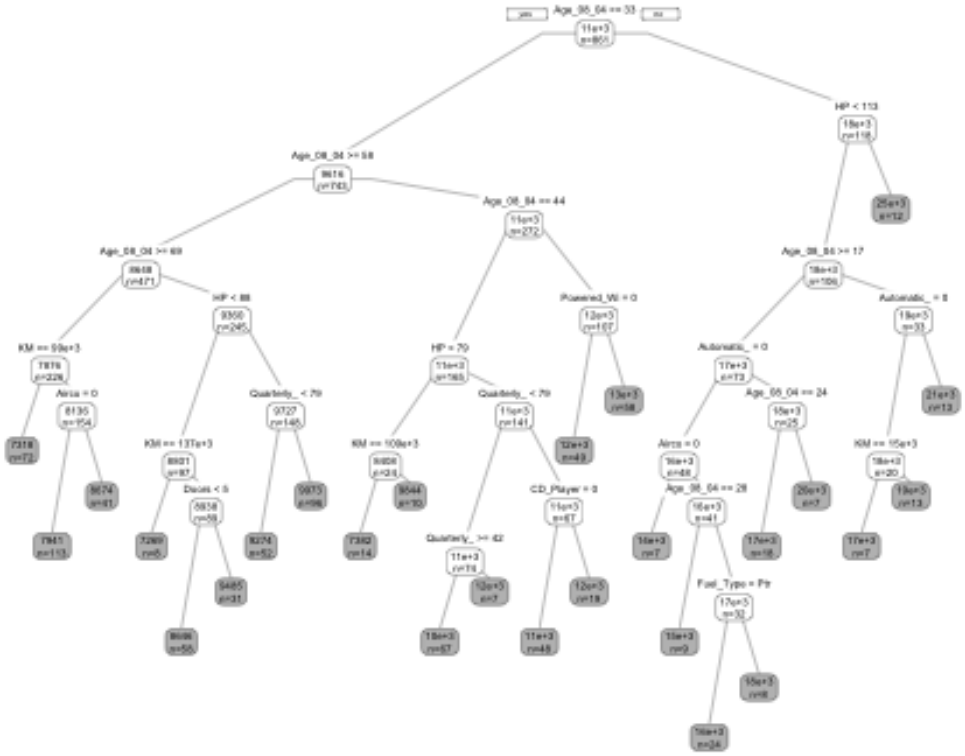
```
# Predicting Prices of Used Cars (Regression Trees)
# I will use the Toyota Corolla data set
# to predict the price of a used Corolla based on it's specifications.

# Load packages and libraries
library(rpart)
library(rpart.plot)
# Read Data
toyota.corolla.df <- read.csv("ToyotaCorolla(1).csv")
# Split data into training (60%) and validation (40%)
# Partition
set.seed(1)
train.index <- sample(c(1:dim(toyota.corolla.df)[1]),
dim(toyota.corolla.df)[1]*0.6)
train.df <- toyota.corolla.df[train.index, ]
valid.df <- toyota.corolla.df[-train.index, ]

# Run a regression tree
# Minimum number of records in a terminal node to 1
# Maximum number of tree levels to 100
# Run least restrictive to 0.001
rt <- rpart(Price ~ Age_08_04 + KM + Fuel_Type + HP + Automatic + Doors +
Quarterly_Tax +
              Mfr_Guarantee + Guarantee_Period + Airco + Automatic_airco +
CD_Player +
              Powered_Windows + Sport_Model + Tow_Bar, data = train.df,
            control = rpart.control(ntree = 100, nodesize = 1, cp = 0.001))

# Print the table
printcp(rt)

##
## Regression tree:
## rpart(formula = Price ~ Age_08_04 + KM + Fuel_Type + HP + Automatic +
##     Doors + Quarterly_Tax + Mfr_Guarantee + Guarantee_Period +
##     Airco + Automatic_airco + CD_Player + Powered_Windows + Sport_Model +
##     Tow_Bar, data = train.df, control = rpart.control(ntree = 100,
##     nodesize = 1, cp = 0.001))
##
## Variables actually used in tree construction:
##  [1] Age_08_04       Airco           Automatic_airco CD_Player
##  [5] Doors           Fuel_Type       HP              KM
##  [9] Powered_Windows Quarterly_Tax
```

```
##
## Root node error: 1.1995e+10/861 = 13931635
##
## n= 861
##
##           CP nsplit rel error  xerror     xstd
## 1  0.6477585      0  1.000000 1.00373 0.086172
## 2  0.1004752      1  0.352242 0.35537 0.030332
## 3  0.0472482      2  0.251766 0.26129 0.029564
## 4  0.0216014      3  0.204518 0.24170 0.026601
## 5  0.0184141      4  0.182917 0.19753 0.016570
## 6  0.0135398      5  0.164503 0.19070 0.016619
## 7  0.0109054      6  0.150963 0.17129 0.015337
## 8  0.0054663      7  0.140057 0.15718 0.014298
## 9  0.0048449      8  0.134591 0.15365 0.013900
## 10 0.0041933      9  0.129746 0.15189 0.013860
## 11 0.0029918     10  0.125553 0.14501 0.013397
## 12 0.0029474     11  0.122561 0.14493 0.013507
## 13 0.0027391     12  0.119614 0.14388 0.013498
## 14 0.0023667     13  0.116875 0.14182 0.013504
## 15 0.0023328     15  0.112141 0.14074 0.013486
## 16 0.0022004     16  0.109808 0.14033 0.013490
## 17 0.0017055     17  0.107608 0.13827 0.013356
## 18 0.0013738     18  0.105903 0.13677 0.013330
## 19 0.0013477     19  0.104529 0.13641 0.013332
## 20 0.0013449     20  0.103181 0.13553 0.013325
## 21 0.0012905     21  0.101836 0.13581 0.013355
## 22 0.0011953     22  0.100546 0.13516 0.013359
## 23 0.0011845     23  0.099350 0.13485 0.013344
## 24 0.0010801     24  0.098166 0.13300 0.013333
## 25 0.0010000     25  0.097086 0.13180 0.013395
```

```
# Prune by lower CP
pruned.rt <- prune(rt, cp =
rt$cptable[which.min(rt$cptable[,"xerror"]),"CP"])
length(pruned.rt$frame$var[pruned.rt$frame$var == "<leaf>"])
```

```
## [1] 26
```

```
# Plot tree
prp(pruned.rt, type = 1, extra = 1, split.font = 1, varlen = -10,
    box.col=ifelse(pruned.rt$frame$var == "<leaf>", 'gray', 'white'))
```



```
# Make predictions
# 3 most important car specifications for predicting the car's price are
Age_08_04, HP, KM
# Define a used car specifications
used.car <- data.frame(Age_08_04 = 25, HP = 110, KM = 68795, Fuel_Type =
'Petrol', Automatic = 1,
                       Doors = 4, Quarterly_Tax =100, Mfr_Guarantee =0,
Guarantee_Period =3, Airco =1,
                       Automatic_airco =1, CD_Player =0, Powered_Windows =1,
Sport_Model =0, Tow_Bar =1)

# Use pruned tree to predict car price
predict(pruned.rt, newdata=used.car)

##     1
## 17358

# Training data has less errors than validation data.
# Predictive performance of valid is worse than training data.
# Due to the partitioned and if the valid data is larger than the train we
will see an unfit data.

# RMS error
```

```r
train.er <- predict(rt, train.df) - train.df$Price
valid.er <- predict(rt, valid.df) - valid.df$Price
# Data frame for Box plot
A =c(train.er, valid.er)
B <- c(rep("Training", length(861)), rep("Validation", length(575)))
rms.er <- data.frame(A, B)
# Box Plot
boxplot(A ~ B, data = rms.er, main="RMSE", xlab = "Data", ylab = "Error",
col=(c("gold","darkgreen")))
```