Jin Kweon - lab 12 (3032235207)

Jin Kweon

11/17/2017

Decision Trees

Q. So, when I output the "plot(tree_carseats)," then does the condition on the parents node true is on the left, and false on the right? ==> True!!!

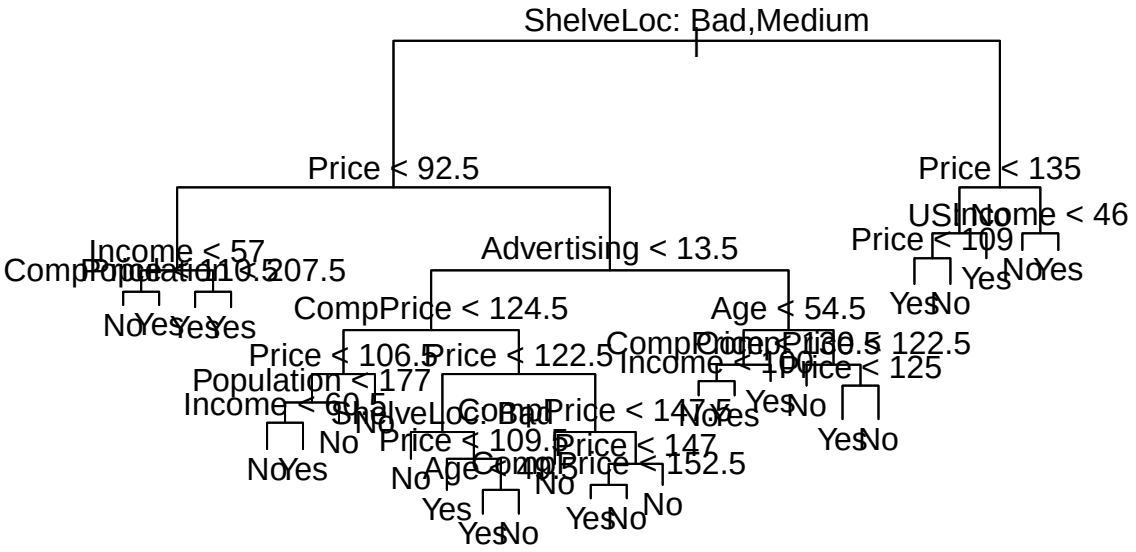
Q. Why do I see error message when I run library(ParallelForest)??? ==> no worry.

```
attach(Carseats)
High <- ifelse(Sales <= 8, "No", "Yes")
carseats <- data.frame(Carseats, High)

tree_carseats <- tree(High ~.-Sales, data = carseats)
summary(tree_carseats)
```

```
##
## Classification tree:
## tree(formula = High ~ . - Sales, data = carseats)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Income"      "CompPrice"   "Population"
## [6] "Advertising" "Age"         "US"
## Number of terminal nodes:  27
## Residual mean deviance:  0.4575 = 170.7 / 373
## Misclassification error rate: 0.09 = 36 / 400
```

```
plot(tree_carseats)
text(tree_carseats, pretty=0)
```




```
tree_carseats
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
```

```
##
##   1) root 400 541.500 No ( 0.59000 0.41000 )
##     2) ShelveLoc: Bad,Medium 315 390.600 No ( 0.68889 0.31111 )
##       4) Price < 92.5 46  56.530 Yes ( 0.30435 0.69565 )
##         8) Income < 57 10  12.220 No ( 0.70000 0.30000 )
##          16) CompPrice < 110.5 5   0.000 No ( 1.00000 0.00000 ) *
##          17) CompPrice > 110.5 5   6.730 Yes ( 0.40000 0.60000 ) *
##         9) Income > 57 36  35.470 Yes ( 0.19444 0.80556 )
##          18) Population < 207.5 16  21.170 Yes ( 0.37500 0.62500 ) *
##          19) Population > 207.5 20   7.941 Yes ( 0.05000 0.95000 ) *
##       5) Price > 92.5 269 299.800 No ( 0.75465 0.24535 )
##        10) Advertising < 13.5 224 213.200 No ( 0.81696 0.18304 )
##          20) CompPrice < 124.5 96  44.890 No ( 0.93750 0.06250 )
##            40) Price < 106.5 38  33.150 No ( 0.84211 0.15789 )
##              80) Population < 177 12  16.300 No ( 0.58333 0.41667 )
##               160) Income < 60.5 6   0.000 No ( 1.00000 0.00000 ) *
##               161) Income > 60.5 6   5.407 Yes ( 0.16667 0.83333 ) *
##              81) Population > 177 26   8.477 No ( 0.96154 0.03846 ) *
##            41) Price > 106.5 58   0.000 No ( 1.00000 0.00000 ) *
##          21) CompPrice > 124.5 128 150.200 No ( 0.72656 0.27344 )
##            42) Price < 122.5 51  70.680 Yes ( 0.49020 0.50980 )
##              84) ShelveLoc: Bad 11   6.702 No ( 0.90909 0.09091 ) *
##              85) ShelveLoc: Medium 40  52.930 Yes ( 0.37500 0.62500 )
##               170) Price < 109.5 16   7.481 Yes ( 0.06250 0.93750 ) *
##               171) Price > 109.5 24  32.600 No ( 0.58333 0.41667 )
##                 342) Age < 49.5 13  16.050 Yes ( 0.30769 0.69231 ) *
##                 343) Age > 49.5 11   6.702 No ( 0.90909 0.09091 ) *
##            43) Price > 122.5 77  55.540 No ( 0.88312 0.11688 )
##              86) CompPrice < 147.5 58  17.400 No ( 0.96552 0.03448 ) *
##              87) CompPrice > 147.5 19  25.010 No ( 0.63158 0.36842 )
##               174) Price < 147 12  16.300 Yes ( 0.41667 0.58333 )
##                 348) CompPrice < 152.5 7   5.742 Yes ( 0.14286 0.85714 ) *
##                 349) CompPrice > 152.5 5   5.004 No ( 0.80000 0.20000 ) *
##               175) Price > 147 7   0.000 No ( 1.00000 0.00000 ) *
##        11) Advertising > 13.5 45  61.830 Yes ( 0.44444 0.55556 )
##          22) Age < 54.5 25  25.020 Yes ( 0.20000 0.80000 )
##            44) CompPrice < 130.5 14  18.250 Yes ( 0.35714 0.64286 )
##              88) Income < 100 9  12.370 No ( 0.55556 0.44444 ) *
##              89) Income > 100 5   0.000 Yes ( 0.00000 1.00000 ) *
##            45) CompPrice > 130.5 11   0.000 Yes ( 0.00000 1.00000 ) *
##          23) Age > 54.5 20  22.490 No ( 0.75000 0.25000 )
##            46) CompPrice < 122.5 10   0.000 No ( 1.00000 0.00000 ) *
##            47) CompPrice > 122.5 10  13.860 No ( 0.50000 0.50000 )
##              94) Price < 125 5   0.000 Yes ( 0.00000 1.00000 ) *
##              95) Price > 125 5   0.000 No ( 1.00000 0.00000 ) *
##     3) ShelveLoc: Good 85  90.330 Yes ( 0.22353 0.77647 )
##       6) Price < 135 68  49.260 Yes ( 0.11765 0.88235 )
##        12) US: No 17  22.070 Yes ( 0.35294 0.64706 )
##          24) Price < 109 8   0.000 Yes ( 0.00000 1.00000 ) *
##          25) Price > 109 9  11.460 No ( 0.66667 0.33333 ) *
##        13) US: Yes 51  16.880 Yes ( 0.03922 0.96078 ) *
##       7) Price > 135 17  22.070 No ( 0.64706 0.35294 )
##        14) Income < 46 6   0.000 No ( 1.00000 0.00000 ) *
##        15) Income > 46 11  15.160 Yes ( 0.45455 0.54545 ) *
```

Random Forests

Q. What does it mean by "Random forests are considered one of the best "off-the-shelf" classifiers with minimal tuning."?? ==> There is no tuning parameter, but you do not need to tune a lot. The default parameter already did good job.

Q. What is really "importance = T"??

Q. The textbook says "In the case of a classification tree, the argument type="class" instructs R to return the actual class prediction." Do I need to include type = class for my exercise? But seems like I got the same result regardless I include this or not...

Q. So, I guess the OOB error rate is the mean prediction error in the training data set? But, why do we really need to get training error rate...? And, I want to get some intution of rf$err.rate...?

Q. Why my importance() function gives different outputs with the one in pg.330??? And, can you help me interpret it?

Q. Why are we doing variable selection here though... I dont really understand.....

```
set.seed(10)
train <- as.vector(createDataPartition(carseats[,2], p = 0.8))[[1]]
rf <- randomForest(High ~ .-Sales, data = carseats[train, ], importance = T)
rf
```

```
##
## Call:
##  randomForest(formula = High ~ . - Sales, data = carseats[train,      ], importance = T)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 19.69%
## Confusion matrix:
##      No Yes class.error
## No  166  23   0.1216931
## Yes  40  91   0.3053435
```

```
rf$err.rate
```

```
##              OOB        No       Yes
##   [1,] 0.3442623 0.3150685 0.3877551
##   [2,] 0.3092784 0.2678571 0.3658537
##   [3,] 0.3020408 0.2620690 0.3600000
##   [4,] 0.2977941 0.2562500 0.3571429
##   [5,] 0.2808219 0.2011834 0.3902439
##   [6,] 0.2662338 0.1944444 0.3671875
##   [7,] 0.2683706 0.1902174 0.3798450
##   [8,] 0.2666667 0.1945946 0.3692308
##   [9,] 0.2807571 0.1818182 0.4230769
##  [10,] 0.2704403 0.1861702 0.3923077
##  [11,] 0.2735849 0.1968085 0.3846154
##  [12,] 0.2515723 0.1861702 0.3461538
##  [13,] 0.2389937 0.1702128 0.3384615
##  [14,] 0.2327044 0.1648936 0.3307692
##  [15,] 0.2351097 0.1587302 0.3461538
##  [16,] 0.2437500 0.1746032 0.3435115
##  [17,] 0.2343750 0.1693122 0.3282443
```

```
##  [18,] 0.2375000 0.1640212 0.3435115
##  [19,] 0.2312500 0.1534392 0.3435115
##  [20,] 0.2312500 0.1481481 0.3511450
##  [21,] 0.2375000 0.1534392 0.3587786
##  [22,] 0.2375000 0.1587302 0.3511450
##  [23,] 0.2343750 0.1481481 0.3587786
##  [24,] 0.2406250 0.1534392 0.3664122
##  [25,] 0.2312500 0.1428571 0.3587786
##  [26,] 0.2375000 0.1534392 0.3587786
##  [27,] 0.2406250 0.1693122 0.3435115
##  [28,] 0.2375000 0.1587302 0.3511450
##  [29,] 0.2406250 0.1587302 0.3587786
##  [30,] 0.2406250 0.1587302 0.3587786
##  [31,] 0.2343750 0.1481481 0.3587786
##  [32,] 0.2406250 0.1587302 0.3587786
##  [33,] 0.2250000 0.1428571 0.3435115
##  [34,] 0.2218750 0.1375661 0.3435115
##  [35,] 0.2125000 0.1269841 0.3358779
##  [36,] 0.2281250 0.1481481 0.3435115
##  [37,] 0.2218750 0.1481481 0.3282443
##  [38,] 0.2218750 0.1428571 0.3358779
##  [39,] 0.2218750 0.1534392 0.3206107
##  [40,] 0.2187500 0.1534392 0.3129771
##  [41,] 0.2250000 0.1481481 0.3358779
##  [42,] 0.2218750 0.1375661 0.3435115
##  [43,] 0.2187500 0.1375661 0.3358779
##  [44,] 0.2281250 0.1428571 0.3511450
##  [45,] 0.2312500 0.1587302 0.3358779
##  [46,] 0.2281250 0.1534392 0.3358779
##  [47,] 0.2375000 0.1587302 0.3511450
##  [48,] 0.2312500 0.1534392 0.3435115
##  [49,] 0.2187500 0.1481481 0.3206107
##  [50,] 0.2312500 0.1534392 0.3435115
##  [51,] 0.2187500 0.1428571 0.3282443
##  [52,] 0.2343750 0.1746032 0.3206107
##  [53,] 0.2218750 0.1481481 0.3282443
##  [54,] 0.2343750 0.1693122 0.3282443
##  [55,] 0.2187500 0.1428571 0.3282443
##  [56,] 0.2250000 0.1534392 0.3282443
##  [57,] 0.2218750 0.1481481 0.3282443
##  [58,] 0.2156250 0.1481481 0.3129771
##  [59,] 0.2250000 0.1587302 0.3206107
##  [60,] 0.2187500 0.1534392 0.3129771
##  [61,] 0.2250000 0.1534392 0.3282443
##  [62,] 0.2156250 0.1428571 0.3206107
##  [63,] 0.2187500 0.1534392 0.3129771
##  [64,] 0.2218750 0.1481481 0.3282443
##  [65,] 0.2281250 0.1587302 0.3282443
##  [66,] 0.2250000 0.1640212 0.3129771
##  [67,] 0.2218750 0.1534392 0.3206107
##  [68,] 0.2187500 0.1587302 0.3053435
##  [69,] 0.2218750 0.1587302 0.3129771
##  [70,] 0.2218750 0.1587302 0.3129771
##  [71,] 0.2218750 0.1534392 0.3206107
```

```
##  [72,] 0.2281250 0.1587302 0.3282443
##  [73,] 0.2281250 0.1587302 0.3282443
##  [74,] 0.2281250 0.1640212 0.3206107
##  [75,] 0.2375000 0.1746032 0.3282443
##  [76,] 0.2250000 0.1534392 0.3282443
##  [77,] 0.2250000 0.1534392 0.3282443
##  [78,] 0.2218750 0.1481481 0.3282443
##  [79,] 0.2218750 0.1428571 0.3358779
##  [80,] 0.2281250 0.1534392 0.3358779
##  [81,] 0.2218750 0.1428571 0.3358779
##  [82,] 0.2187500 0.1428571 0.3282443
##  [83,] 0.2156250 0.1428571 0.3206107
##  [84,] 0.2187500 0.1428571 0.3282443
##  [85,] 0.2093750 0.1375661 0.3129771
##  [86,] 0.2093750 0.1428571 0.3053435
##  [87,] 0.2125000 0.1428571 0.3129771
##  [88,] 0.2187500 0.1481481 0.3206107
##  [89,] 0.2156250 0.1375661 0.3282443
##  [90,] 0.2156250 0.1428571 0.3206107
##  [91,] 0.2062500 0.1428571 0.2977099
##  [92,] 0.2125000 0.1375661 0.3206107
##  [93,] 0.2156250 0.1375661 0.3282443
##  [94,] 0.2218750 0.1534392 0.3206107
##  [95,] 0.2062500 0.1428571 0.2977099
##  [96,] 0.2093750 0.1428571 0.3053435
##  [97,] 0.2156250 0.1481481 0.3129771
##  [98,] 0.2062500 0.1375661 0.3053435
##  [99,] 0.2125000 0.1375661 0.3206107
## [100,] 0.2093750 0.1375661 0.3129771
## [101,] 0.2125000 0.1428571 0.3129771
## [102,] 0.2093750 0.1428571 0.3053435
## [103,] 0.2093750 0.1428571 0.3053435
## [104,] 0.2125000 0.1428571 0.3129771
## [105,] 0.2093750 0.1428571 0.3053435
## [106,] 0.2125000 0.1428571 0.3129771
## [107,] 0.2093750 0.1428571 0.3053435
## [108,] 0.2062500 0.1375661 0.3053435
## [109,] 0.2062500 0.1322751 0.3129771
## [110,] 0.2062500 0.1322751 0.3129771
## [111,] 0.2000000 0.1269841 0.3053435
## [112,] 0.2000000 0.1269841 0.3053435
## [113,] 0.2031250 0.1269841 0.3129771
## [114,] 0.2031250 0.1269841 0.3129771
## [115,] 0.2062500 0.1322751 0.3129771
## [116,] 0.2031250 0.1269841 0.3129771
## [117,] 0.2000000 0.1269841 0.3053435
## [118,] 0.2000000 0.1269841 0.3053435
## [119,] 0.2031250 0.1322751 0.3053435
## [120,] 0.1968750 0.1269841 0.2977099
## [121,] 0.2000000 0.1269841 0.3053435
## [122,] 0.1968750 0.1216931 0.3053435
## [123,] 0.2031250 0.1322751 0.3053435
## [124,] 0.2031250 0.1269841 0.3129771
## [125,] 0.2000000 0.1216931 0.3129771
```

```
## [126,] 0.2000000 0.1216931 0.3129771
## [127,] 0.2000000 0.1269841 0.3053435
## [128,] 0.1968750 0.1216931 0.3053435
## [129,] 0.1968750 0.1216931 0.3053435
## [130,] 0.1968750 0.1216931 0.3053435
## [131,] 0.1937500 0.1216931 0.2977099
## [132,] 0.1968750 0.1269841 0.2977099
## [133,] 0.2031250 0.1375661 0.2977099
## [134,] 0.2000000 0.1322751 0.2977099
## [135,] 0.2031250 0.1322751 0.3053435
## [136,] 0.2031250 0.1375661 0.2977099
## [137,] 0.2031250 0.1322751 0.3053435
## [138,] 0.2031250 0.1375661 0.2977099
## [139,] 0.2062500 0.1375661 0.3053435
## [140,] 0.2000000 0.1269841 0.3053435
## [141,] 0.1937500 0.1269841 0.2900763
## [142,] 0.2000000 0.1269841 0.3053435
## [143,] 0.1968750 0.1216931 0.3053435
## [144,] 0.1968750 0.1164021 0.3129771
## [145,] 0.1968750 0.1164021 0.3129771
## [146,] 0.1968750 0.1216931 0.3053435
## [147,] 0.1968750 0.1269841 0.2977099
## [148,] 0.1906250 0.1164021 0.2977099
## [149,] 0.2000000 0.1322751 0.2977099
## [150,] 0.1937500 0.1164021 0.3053435
## [151,] 0.1968750 0.1216931 0.3053435
## [152,] 0.1937500 0.1164021 0.3053435
## [153,] 0.1937500 0.1216931 0.2977099
## [154,] 0.1968750 0.1216931 0.3053435
## [155,] 0.1968750 0.1216931 0.3053435
## [156,] 0.1937500 0.1164021 0.3053435
## [157,] 0.1937500 0.1164021 0.3053435
## [158,] 0.1937500 0.1164021 0.3053435
## [159,] 0.2000000 0.1216931 0.3129771
## [160,] 0.2000000 0.1216931 0.3129771
## [161,] 0.1968750 0.1216931 0.3053435
## [162,] 0.1937500 0.1164021 0.3053435
## [163,] 0.2000000 0.1269841 0.3053435
## [164,] 0.1968750 0.1216931 0.3053435
## [165,] 0.1937500 0.1216931 0.2977099
## [166,] 0.2031250 0.1269841 0.3129771
## [167,] 0.2000000 0.1216931 0.3129771
## [168,] 0.2000000 0.1216931 0.3129771
## [169,] 0.1968750 0.1164021 0.3129771
## [170,] 0.1937500 0.1216931 0.2977099
## [171,] 0.1968750 0.1216931 0.3053435
## [172,] 0.1968750 0.1216931 0.3053435
## [173,] 0.1968750 0.1164021 0.3129771
## [174,] 0.1937500 0.1164021 0.3053435
## [175,] 0.1937500 0.1216931 0.2977099
## [176,] 0.2000000 0.1269841 0.3053435
## [177,] 0.1968750 0.1216931 0.3053435
## [178,] 0.1937500 0.1216931 0.2977099
## [179,] 0.1937500 0.1216931 0.2977099
```

```
## [180,] 0.1937500 0.1216931 0.2977099
## [181,] 0.1906250 0.1216931 0.2900763
## [182,] 0.1906250 0.1216931 0.2900763
## [183,] 0.1906250 0.1216931 0.2900763
## [184,] 0.1937500 0.1216931 0.2977099
## [185,] 0.1937500 0.1216931 0.2977099
## [186,] 0.1906250 0.1216931 0.2900763
## [187,] 0.1906250 0.1164021 0.2977099
## [188,] 0.1937500 0.1216931 0.2977099
## [189,] 0.2000000 0.1269841 0.3053435
## [190,] 0.2000000 0.1269841 0.3053435
## [191,] 0.2000000 0.1269841 0.3053435
## [192,] 0.2000000 0.1269841 0.3053435
## [193,] 0.1968750 0.1216931 0.3053435
## [194,] 0.1968750 0.1216931 0.3053435
## [195,] 0.2000000 0.1269841 0.3053435
## [196,] 0.1937500 0.1216931 0.2977099
## [197,] 0.2031250 0.1269841 0.3129771
## [198,] 0.2031250 0.1269841 0.3129771
## [199,] 0.2000000 0.1216931 0.3129771
## [200,] 0.1968750 0.1216931 0.3053435
## [201,] 0.1968750 0.1164021 0.3129771
## [202,] 0.1937500 0.1111111 0.3129771
## [203,] 0.1937500 0.1111111 0.3129771
## [204,] 0.1937500 0.1164021 0.3053435
## [205,] 0.2031250 0.1269841 0.3129771
## [206,] 0.2031250 0.1216931 0.3206107
## [207,] 0.2031250 0.1269841 0.3129771
## [208,] 0.2000000 0.1216931 0.3129771
## [209,] 0.2000000 0.1216931 0.3129771
## [210,] 0.2000000 0.1216931 0.3129771
## [211,] 0.1968750 0.1216931 0.3053435
## [212,] 0.2031250 0.1322751 0.3053435
## [213,] 0.2031250 0.1322751 0.3053435
## [214,] 0.2000000 0.1216931 0.3129771
## [215,] 0.2062500 0.1322751 0.3129771
## [216,] 0.1968750 0.1216931 0.3053435
## [217,] 0.1968750 0.1216931 0.3053435
## [218,] 0.2031250 0.1269841 0.3129771
## [219,] 0.2031250 0.1269841 0.3129771
## [220,] 0.2031250 0.1269841 0.3129771
## [221,] 0.2000000 0.1216931 0.3129771
## [222,] 0.2031250 0.1269841 0.3129771
## [223,] 0.1968750 0.1216931 0.3053435
## [224,] 0.2000000 0.1269841 0.3053435
## [225,] 0.2031250 0.1269841 0.3129771
## [226,] 0.1968750 0.1216931 0.3053435
## [227,] 0.1968750 0.1216931 0.3053435
## [228,] 0.2031250 0.1322751 0.3053435
## [229,] 0.2000000 0.1269841 0.3053435
## [230,] 0.2000000 0.1216931 0.3129771
## [231,] 0.1968750 0.1216931 0.3053435
## [232,] 0.2000000 0.1216931 0.3129771
## [233,] 0.2000000 0.1216931 0.3129771
```

```
## [234,] 0.2000000 0.1216931 0.3129771
## [235,] 0.1968750 0.1216931 0.3053435
## [236,] 0.1968750 0.1164021 0.3129771
## [237,] 0.1968750 0.1164021 0.3129771
## [238,] 0.2000000 0.1216931 0.3129771
## [239,] 0.1968750 0.1216931 0.3053435
## [240,] 0.1968750 0.1216931 0.3053435
## [241,] 0.1968750 0.1216931 0.3053435
## [242,] 0.2000000 0.1216931 0.3129771
## [243,] 0.1937500 0.1164021 0.3053435
## [244,] 0.1968750 0.1216931 0.3053435
## [245,] 0.1968750 0.1269841 0.2977099
## [246,] 0.1968750 0.1269841 0.2977099
## [247,] 0.1937500 0.1216931 0.2977099
## [248,] 0.1937500 0.1216931 0.2977099
## [249,] 0.1968750 0.1269841 0.2977099
## [250,] 0.1937500 0.1216931 0.2977099
## [251,] 0.1968750 0.1216931 0.3053435
## [252,] 0.1968750 0.1216931 0.3053435
## [253,] 0.1937500 0.1216931 0.2977099
## [254,] 0.1968750 0.1216931 0.3053435
## [255,] 0.1968750 0.1216931 0.3053435
## [256,] 0.2000000 0.1216931 0.3129771
## [257,] 0.2000000 0.1216931 0.3129771
## [258,] 0.1937500 0.1216931 0.2977099
## [259,] 0.1968750 0.1216931 0.3053435
## [260,] 0.1937500 0.1216931 0.2977099
## [261,] 0.1937500 0.1164021 0.3053435
## [262,] 0.1937500 0.1216931 0.2977099
## [263,] 0.1937500 0.1164021 0.3053435
## [264,] 0.1937500 0.1164021 0.3053435
## [265,] 0.1937500 0.1164021 0.3053435
## [266,] 0.1906250 0.1164021 0.2977099
## [267,] 0.1906250 0.1164021 0.2977099
## [268,] 0.1937500 0.1164021 0.3053435
## [269,] 0.1906250 0.1164021 0.2977099
## [270,] 0.1906250 0.1164021 0.2977099
## [271,] 0.1906250 0.1164021 0.2977099
## [272,] 0.1906250 0.1164021 0.2977099
## [273,] 0.1906250 0.1164021 0.2977099
## [274,] 0.1937500 0.1164021 0.3053435
## [275,] 0.1906250 0.1164021 0.2977099
## [276,] 0.1906250 0.1164021 0.2977099
## [277,] 0.1937500 0.1164021 0.3053435
## [278,] 0.1906250 0.1164021 0.2977099
## [279,] 0.1937500 0.1164021 0.3053435
## [280,] 0.1906250 0.1164021 0.2977099
## [281,] 0.1906250 0.1164021 0.2977099
## [282,] 0.1906250 0.1164021 0.2977099
## [283,] 0.1875000 0.1164021 0.2900763
## [284,] 0.1875000 0.1164021 0.2900763
## [285,] 0.1875000 0.1164021 0.2900763
## [286,] 0.1875000 0.1164021 0.2900763
## [287,] 0.1906250 0.1164021 0.2977099
```

```
## [288,] 0.1875000 0.1164021 0.2900763
## [289,] 0.1875000 0.1164021 0.2900763
## [290,] 0.1875000 0.1164021 0.2900763
## [291,] 0.1906250 0.1164021 0.2977099
## [292,] 0.1875000 0.1164021 0.2900763
## [293,] 0.1875000 0.1164021 0.2900763
## [294,] 0.1875000 0.1164021 0.2900763
## [295,] 0.1906250 0.1164021 0.2977099
## [296,] 0.1875000 0.1164021 0.2900763
## [297,] 0.1875000 0.1164021 0.2900763
## [298,] 0.1906250 0.1164021 0.2977099
## [299,] 0.1875000 0.1164021 0.2900763
## [300,] 0.1937500 0.1164021 0.3053435
## [301,] 0.1875000 0.1164021 0.2900763
## [302,] 0.1875000 0.1164021 0.2900763
## [303,] 0.1875000 0.1164021 0.2900763
## [304,] 0.1875000 0.1164021 0.2900763
## [305,] 0.1875000 0.1164021 0.2900763
## [306,] 0.1875000 0.1164021 0.2900763
## [307,] 0.1875000 0.1164021 0.2900763
## [308,] 0.1875000 0.1164021 0.2900763
## [309,] 0.1875000 0.1164021 0.2900763
## [310,] 0.1906250 0.1216931 0.2900763
## [311,] 0.1906250 0.1216931 0.2900763
## [312,] 0.1906250 0.1216931 0.2900763
## [313,] 0.1906250 0.1216931 0.2900763
## [314,] 0.1937500 0.1216931 0.2977099
## [315,] 0.1906250 0.1216931 0.2900763
## [316,] 0.1906250 0.1216931 0.2900763
## [317,] 0.1906250 0.1216931 0.2900763
## [318,] 0.1875000 0.1216931 0.2824427
## [319,] 0.1875000 0.1216931 0.2824427
## [320,] 0.1906250 0.1216931 0.2900763
## [321,] 0.1906250 0.1216931 0.2900763
## [322,] 0.1875000 0.1216931 0.2824427
## [323,] 0.1906250 0.1216931 0.2900763
## [324,] 0.1875000 0.1216931 0.2824427
## [325,] 0.1875000 0.1216931 0.2824427
## [326,] 0.1875000 0.1216931 0.2824427
## [327,] 0.1875000 0.1216931 0.2824427
## [328,] 0.1906250 0.1216931 0.2900763
## [329,] 0.1906250 0.1216931 0.2900763
## [330,] 0.1937500 0.1216931 0.2977099
## [331,] 0.1937500 0.1216931 0.2977099
## [332,] 0.1937500 0.1216931 0.2977099
## [333,] 0.1906250 0.1216931 0.2900763
## [334,] 0.1906250 0.1216931 0.2900763
## [335,] 0.1906250 0.1216931 0.2900763
## [336,] 0.1906250 0.1216931 0.2900763
## [337,] 0.1906250 0.1216931 0.2900763
## [338,] 0.1875000 0.1216931 0.2824427
## [339,] 0.1906250 0.1216931 0.2900763
## [340,] 0.1906250 0.1216931 0.2900763
## [341,] 0.1906250 0.1216931 0.2900763
```

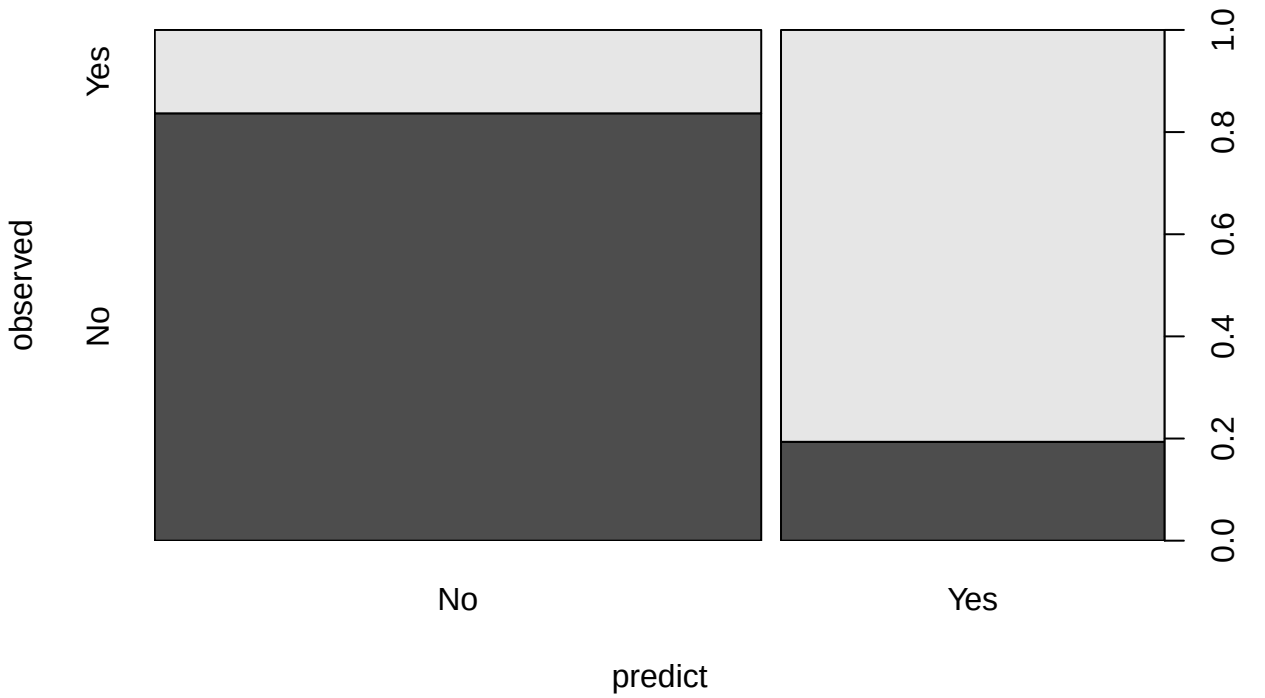
```
## [342,] 0.1875000 0.1164021 0.2900763
## [343,] 0.1906250 0.1164021 0.2977099
## [344,] 0.1906250 0.1164021 0.2977099
## [345,] 0.1906250 0.1164021 0.2977099
## [346,] 0.1937500 0.1216931 0.2977099
## [347,] 0.1906250 0.1164021 0.2977099
## [348,] 0.1906250 0.1216931 0.2900763
## [349,] 0.1906250 0.1216931 0.2900763
## [350,] 0.1906250 0.1216931 0.2900763
## [351,] 0.1875000 0.1164021 0.2900763
## [352,] 0.1875000 0.1164021 0.2900763
## [353,] 0.1906250 0.1164021 0.2977099
## [354,] 0.1875000 0.1164021 0.2900763
## [355,] 0.1906250 0.1216931 0.2900763
## [356,] 0.1906250 0.1164021 0.2977099
## [357,] 0.1875000 0.1164021 0.2900763
## [358,] 0.1906250 0.1164021 0.2977099
## [359,] 0.1906250 0.1164021 0.2977099
## [360,] 0.1875000 0.1164021 0.2900763
## [361,] 0.1875000 0.1164021 0.2900763
## [362,] 0.1937500 0.1216931 0.2977099
## [363,] 0.1937500 0.1216931 0.2977099
## [364,] 0.1937500 0.1216931 0.2977099
## [365,] 0.1937500 0.1216931 0.2977099
## [366,] 0.1906250 0.1164021 0.2977099
## [367,] 0.1906250 0.1164021 0.2977099
## [368,] 0.1906250 0.1164021 0.2977099
## [369,] 0.1906250 0.1164021 0.2977099
## [370,] 0.1906250 0.1164021 0.2977099
## [371,] 0.1906250 0.1164021 0.2977099
## [372,] 0.1906250 0.1164021 0.2977099
## [373,] 0.1906250 0.1164021 0.2977099
## [374,] 0.1875000 0.1164021 0.2900763
## [375,] 0.1906250 0.1164021 0.2977099
## [376,] 0.1906250 0.1164021 0.2977099
## [377,] 0.1906250 0.1164021 0.2977099
## [378,] 0.1906250 0.1164021 0.2977099
## [379,] 0.1906250 0.1164021 0.2977099
## [380,] 0.1875000 0.1164021 0.2900763
## [381,] 0.1906250 0.1164021 0.2977099
## [382,] 0.1875000 0.1164021 0.2900763
## [383,] 0.1875000 0.1164021 0.2900763
## [384,] 0.1875000 0.1164021 0.2900763
## [385,] 0.1843750 0.1164021 0.2824427
## [386,] 0.1875000 0.1164021 0.2900763
## [387,] 0.1875000 0.1164021 0.2900763
## [388,] 0.1843750 0.1164021 0.2824427
## [389,] 0.1875000 0.1164021 0.2900763
## [390,] 0.1906250 0.1164021 0.2977099
## [391,] 0.1906250 0.1164021 0.2977099
## [392,] 0.1906250 0.1164021 0.2977099
## [393,] 0.1906250 0.1164021 0.2977099
## [394,] 0.1906250 0.1164021 0.2977099
## [395,] 0.1937500 0.1164021 0.3053435
```

```
## [396,] 0.1937500 0.1164021 0.3053435
## [397,] 0.1968750 0.1164021 0.3129771
## [398,] 0.1937500 0.1164021 0.3053435
## [399,] 0.1937500 0.1164021 0.3053435
## [400,] 0.1968750 0.1164021 0.3129771
## [401,] 0.1968750 0.1164021 0.3129771
## [402,] 0.1937500 0.1164021 0.3053435
## [403,] 0.1968750 0.1164021 0.3129771
## [404,] 0.1968750 0.1164021 0.3129771
## [405,] 0.1968750 0.1164021 0.3129771
## [406,] 0.1937500 0.1164021 0.3053435
## [407,] 0.1937500 0.1164021 0.3053435
## [408,] 0.1937500 0.1164021 0.3053435
## [409,] 0.1937500 0.1164021 0.3053435
## [410,] 0.1937500 0.1164021 0.3053435
## [411,] 0.1937500 0.1164021 0.3053435
## [412,] 0.1906250 0.1164021 0.2977099
## [413,] 0.1906250 0.1164021 0.2977099
## [414,] 0.1937500 0.1164021 0.3053435
## [415,] 0.1906250 0.1164021 0.2977099
## [416,] 0.1906250 0.1164021 0.2977099
## [417,] 0.1906250 0.1164021 0.2977099
## [418,] 0.1906250 0.1164021 0.2977099
## [419,] 0.1906250 0.1164021 0.2977099
## [420,] 0.1906250 0.1164021 0.2977099
## [421,] 0.1906250 0.1164021 0.2977099
## [422,] 0.1906250 0.1164021 0.2977099
## [423,] 0.1906250 0.1164021 0.2977099
## [424,] 0.1906250 0.1164021 0.2977099
## [425,] 0.1906250 0.1164021 0.2977099
## [426,] 0.1906250 0.1164021 0.2977099
## [427,] 0.1906250 0.1164021 0.2977099
## [428,] 0.1906250 0.1164021 0.2977099
## [429,] 0.1906250 0.1164021 0.2977099
## [430,] 0.1906250 0.1164021 0.2977099
## [431,] 0.1906250 0.1164021 0.2977099
## [432,] 0.1937500 0.1164021 0.3053435
## [433,] 0.1937500 0.1216931 0.2977099
## [434,] 0.1906250 0.1164021 0.2977099
## [435,] 0.1906250 0.1164021 0.2977099
## [436,] 0.1906250 0.1164021 0.2977099
## [437,] 0.1906250 0.1164021 0.2977099
## [438,] 0.1906250 0.1164021 0.2977099
## [439,] 0.1906250 0.1164021 0.2977099
## [440,] 0.1906250 0.1164021 0.2977099
## [441,] 0.1906250 0.1164021 0.2977099
## [442,] 0.1906250 0.1164021 0.2977099
## [443,] 0.1937500 0.1164021 0.3053435
## [444,] 0.1906250 0.1164021 0.2977099
## [445,] 0.1937500 0.1164021 0.3053435
## [446,] 0.1937500 0.1164021 0.3053435
## [447,] 0.1906250 0.1164021 0.2977099
## [448,] 0.1937500 0.1164021 0.3053435
## [449,] 0.1937500 0.1164021 0.3053435
```

```
## [450,] 0.1968750 0.1216931 0.3053435
## [451,] 0.1968750 0.1216931 0.3053435
## [452,] 0.1968750 0.1216931 0.3053435
## [453,] 0.1968750 0.1216931 0.3053435
## [454,] 0.1968750 0.1216931 0.3053435
## [455,] 0.2000000 0.1269841 0.3053435
## [456,] 0.1968750 0.1216931 0.3053435
## [457,] 0.2000000 0.1269841 0.3053435
## [458,] 0.1968750 0.1216931 0.3053435
## [459,] 0.1968750 0.1216931 0.3053435
## [460,] 0.1968750 0.1216931 0.3053435
## [461,] 0.1968750 0.1216931 0.3053435
## [462,] 0.1937500 0.1216931 0.2977099
## [463,] 0.1937500 0.1216931 0.2977099
## [464,] 0.1937500 0.1216931 0.2977099
## [465,] 0.1968750 0.1216931 0.3053435
## [466,] 0.1968750 0.1216931 0.3053435
## [467,] 0.1937500 0.1216931 0.2977099
## [468,] 0.1937500 0.1216931 0.2977099
## [469,] 0.1968750 0.1269841 0.2977099
## [470,] 0.1968750 0.1269841 0.2977099
## [471,] 0.1968750 0.1269841 0.2977099
## [472,] 0.1968750 0.1269841 0.2977099
## [473,] 0.1968750 0.1269841 0.2977099
## [474,] 0.1968750 0.1269841 0.2977099
## [475,] 0.1968750 0.1216931 0.3053435
## [476,] 0.1968750 0.1269841 0.2977099
## [477,] 0.1968750 0.1216931 0.3053435
## [478,] 0.1968750 0.1216931 0.3053435
## [479,] 0.1937500 0.1216931 0.2977099
## [480,] 0.1937500 0.1216931 0.2977099
## [481,] 0.1937500 0.1216931 0.2977099
## [482,] 0.1968750 0.1216931 0.3053435
## [483,] 0.1937500 0.1216931 0.2977099
## [484,] 0.1937500 0.1216931 0.2977099
## [485,] 0.1968750 0.1216931 0.3053435
## [486,] 0.1937500 0.1216931 0.2977099
## [487,] 0.1937500 0.1216931 0.2977099
## [488,] 0.1937500 0.1216931 0.2977099
## [489,] 0.1937500 0.1216931 0.2977099
## [490,] 0.1937500 0.1216931 0.2977099
## [491,] 0.1937500 0.1216931 0.2977099
## [492,] 0.1968750 0.1216931 0.3053435
## [493,] 0.1968750 0.1216931 0.3053435
## [494,] 0.1937500 0.1216931 0.2977099
## [495,] 0.1968750 0.1216931 0.3053435
## [496,] 0.1968750 0.1216931 0.3053435
## [497,] 0.1968750 0.1216931 0.3053435
## [498,] 0.1968750 0.1216931 0.3053435
## [499,] 0.1968750 0.1216931 0.3053435
## [500,] 0.1968750 0.1216931 0.3053435
```

```
yhat <- predict(rf, newdata = carseats[-train, ])
hightest <- carseats[-train, "High"]
```

```
plot(yhat, hightest, xlab = "predict", ylab = "observed")
```




```
confusion <- table(yhat, hightest)
(confusion[1,2] + confusion[2,1]) / (confusion[1,2] + confusion[1,1] + confusion[2,1] + confusion[2,2])
```

```
## [1] 0.175
```

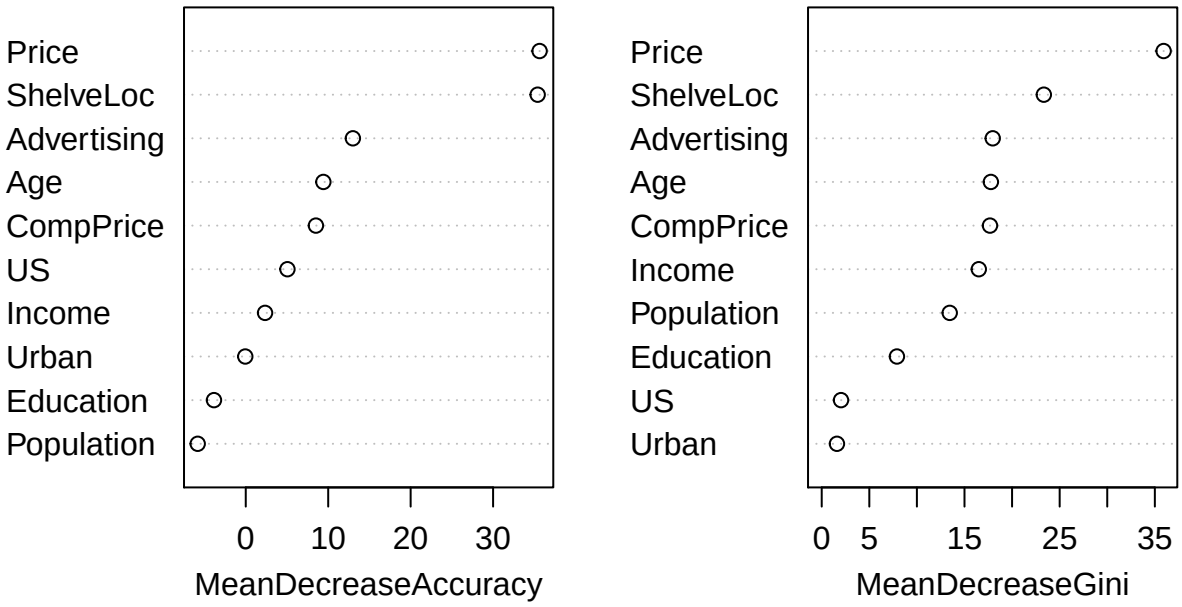
```
mean((as.numeric(yhat) - as.numeric((hightest)))^2) #other way to do it...
```

```
## [1] 0.175
```

```
randomForest::importance(rf)
```

```
##                    No         Yes MeanDecreaseAccuracy MeanDecreaseGini
## CompPrice    8.36916959  2.91117011           8.50864854        17.678930
## Income       0.39579991  3.25532567           2.34027456        16.503003
## Advertising  6.65805187 12.05615582          12.99366314        17.967101
## Population  -2.88848674 -5.07534885          -5.82995892        13.449605
## Price       27.19897167 25.90688901          35.64189722        35.922717
## ShelveLoc   29.48850638 28.53645200          35.40435451        23.335803
## Age          6.07907833  6.68942744           9.41109449        17.773115
## Education   -3.00970397 -2.61744796          -3.85681059         7.902363
## Urban       -0.09313098  0.08359962          -0.05974897         1.598495
## US           0.27545841  5.78679251           5.05518967         2.028199
```

```
varImpPlot(rf)
```



Boosted Trees

Q. What does it mean by "The idea of boosting is to iteratively fit a small tree to the residuals from the current model"???

Q. Dont we have to say "distribution = bernoulli" as this is classification problem. . . ??

Q. How to set 0.5 as the cutoff for the predicted probabilities??

Q. So, what is the difference between boosted trees and random forest. . . ? I still dont understand when they important variables. . . .

Q. When I do predict to get test error rate, do I include gbm inside of for-loop as well???

Q. How do I understand shrinkage parameter conceptually here in tree???

Q. Why my plot shows monotonically increasing? Am I supposed to have a diagram looks quadratic??? I found that if I split data into training and testing like "as.vector(createDataPartition(1:nrow(carseats), p = 0.8))[[1]]" for data partition, then I get monotonically increasing, but if I do "as.vector(createDataPartition(carseats[,1], p = 0.8))[[1]]", then I got quadratic. . . So, I guess data partion makes some difference. What is wrong. . . ??? (please refer to the codes below for evidence. . . .)

Q. Why do I have high test erorr as I increase interaction? Intuively saying, am I supposed to get small mse if the variable is more flexible??

You do not do MSE for classification, but you are doing comparing actual and prediction to get error rates. . . .

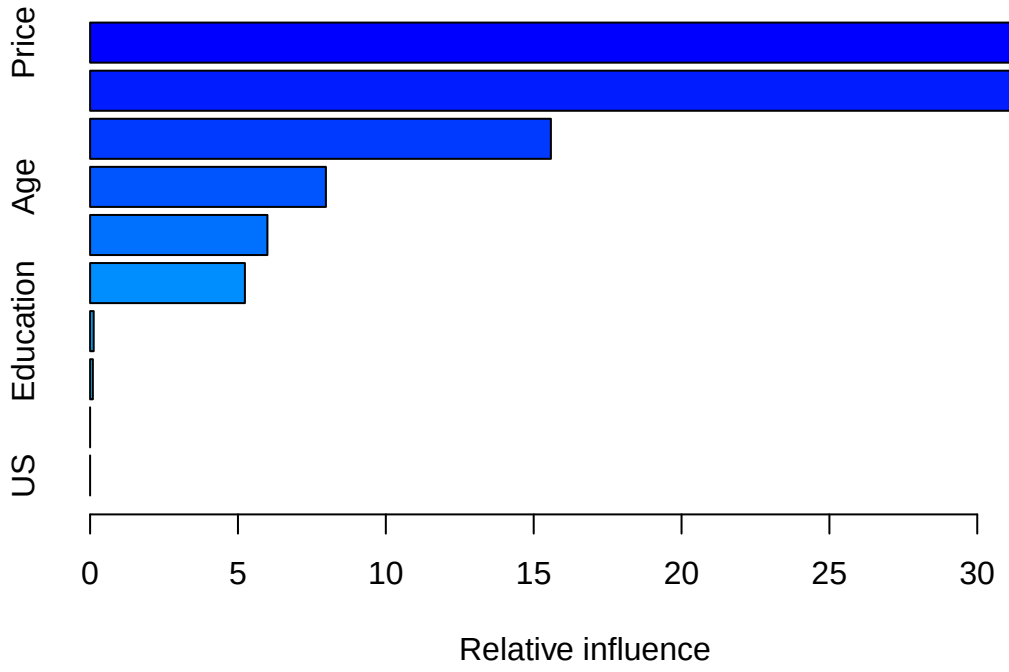
```
carseats$High <- as.numeric(carseats$High) - 1

#1

boosting <- gbm(High~.-Sales, data = carseats[train, ], distribution = "bernoulli", n.trees = 5000,
                interaction.depth = 1)
boosting
```

```
## gbm(formula = High ~ . - Sales, distribution = "bernoulli", data = carseats[train,
##     ], n.trees = 5000, interaction.depth = 1)
## A gradient boosted model with bernoulli loss function.
## 5000 iterations were performed.
## There were 10 predictors of which 8 had non-zero influence.
```

```
summary(boosting)
```

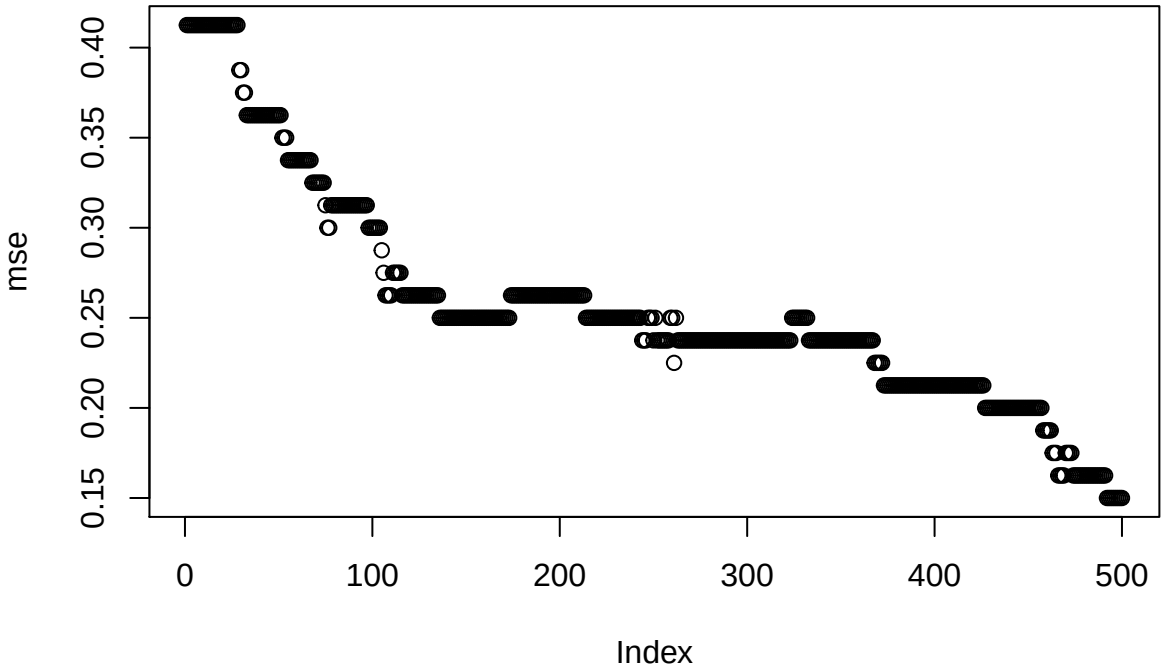



```
##                    var     rel.inf
## Price            Price 33.81445486
## ShelveLoc     ShelveLoc 31.18481213
## Advertising Advertising 15.58206140
## Age                Age  7.97407974
## Income          Income  5.99578098
## CompPrice     CompPrice  5.23490756
## Education     Education  0.12080515
## Population   Population  0.09309819
## Urban             Urban  0.00000000
## US                   US  0.00000000
```

```
mse <- c()
thresh <- 0.5
```

```r
for(i in 1:500){
  #If I do not type = "response", they will give you logit output.
  yhat <- predict(boosting, newdata = carseats[-train, ], n.trees = (10 * i), type = "response")
  yhat <- (yhat > thresh)
  mse[i] <- mean(yhat != carseats$High[-train])
}


plot(mse)
```

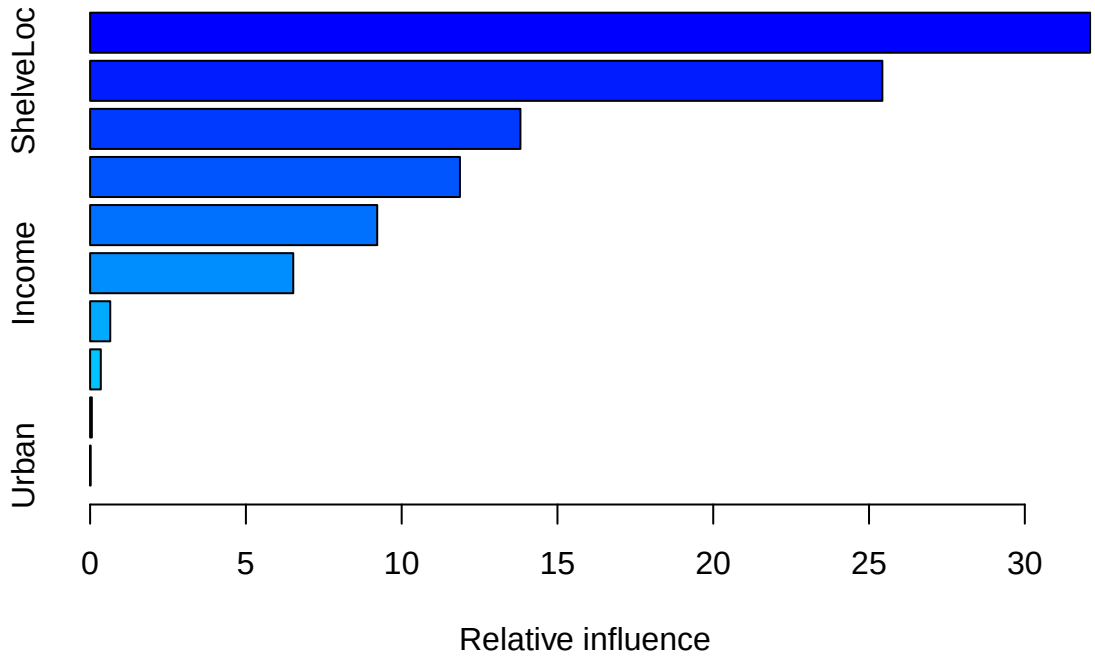



```r
#2

boosting2 <- gbm(High~.-Sales, data = carseats[train, ], distribution = "bernoulli",
                 n.trees = 5000, interaction.depth = 2)
boosting2
```

```
## gbm(formula = High ~ . - Sales, distribution = "bernoulli", data = carseats[train,
##     ], n.trees = 5000, interaction.depth = 2)
## A gradient boosted model with bernoulli loss function.
## 5000 iterations were performed.
## There were 10 predictors of which 10 had non-zero influence.
```

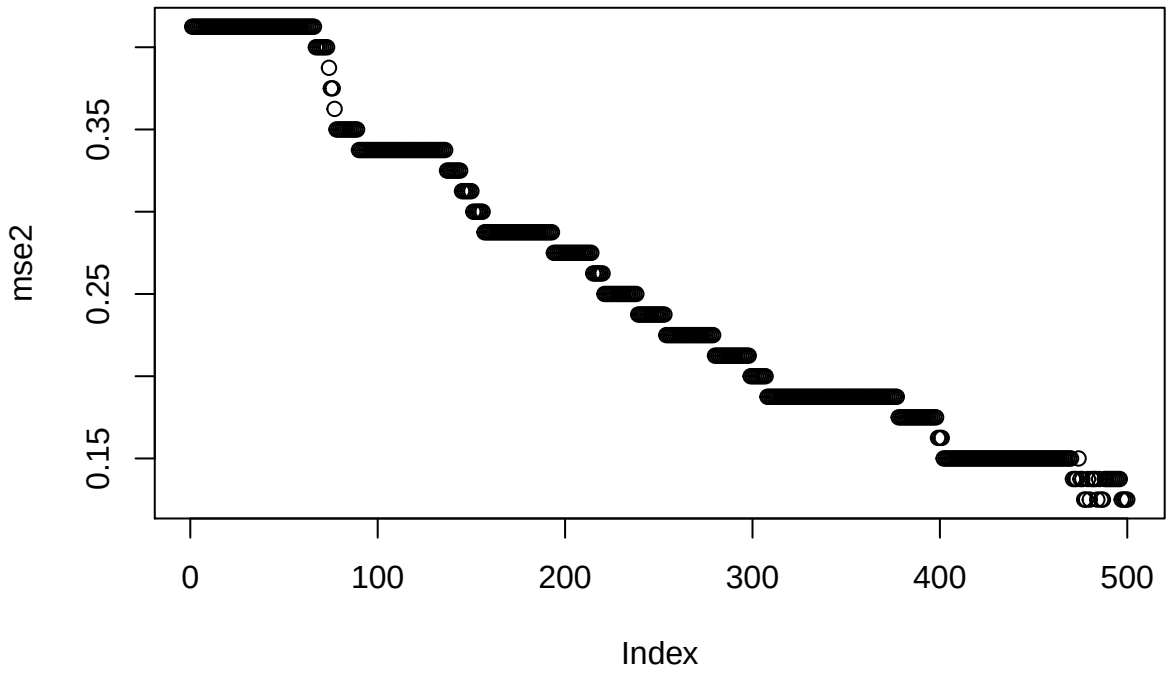
```r
summary(boosting2)
```




```
##                     var     rel.inf
## Price             Price 32.09476062
## ShelveLoc     ShelveLoc 25.42750999
## Advertising Advertising 13.81159526
## CompPrice     CompPrice 11.87025883
## Age                 Age  9.21481597
## Income           Income  6.52005165
## Population   Population  0.64664477
## Education     Education  0.34302613
## US                   US  0.05578035
## Urban             Urban  0.01555645
```

```r
mse2 <- c()
for(i in 1:500){
  yhat <- predict(boosting2, newdata = carseats[-train, ], n.trees = (10 * i))
  yhat <- (yhat > thresh)
  mse2[i] <- mean(yhat != carseats$High[-train])
}


plot(mse2)
```

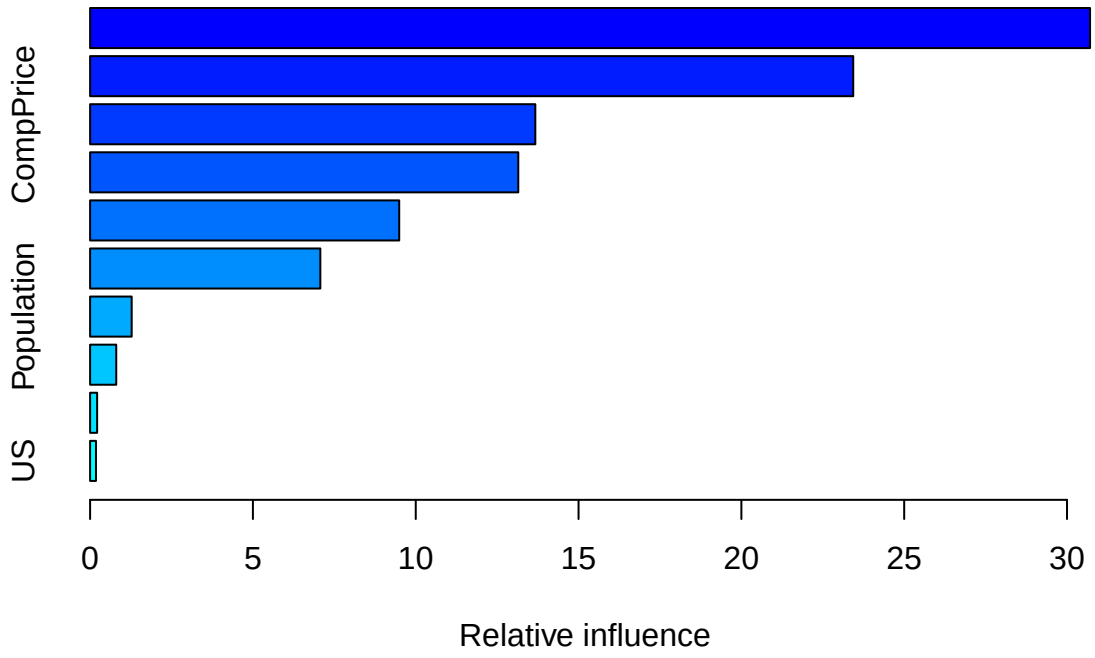



```
#3

boosting3 <- gbm(High~.-Sales, data = carseats[train, ], distribution = "bernoulli",
                 n.trees = 5000, interaction.depth = 3)
boosting3
```

```
## gbm(formula = High ~ . - Sales, distribution = "bernoulli", data = carseats[train,
##     ], n.trees = 5000, interaction.depth = 3)
## A gradient boosted model with bernoulli loss function.
## 5000 iterations were performed.
## There were 10 predictors of which 10 had non-zero influence.
```

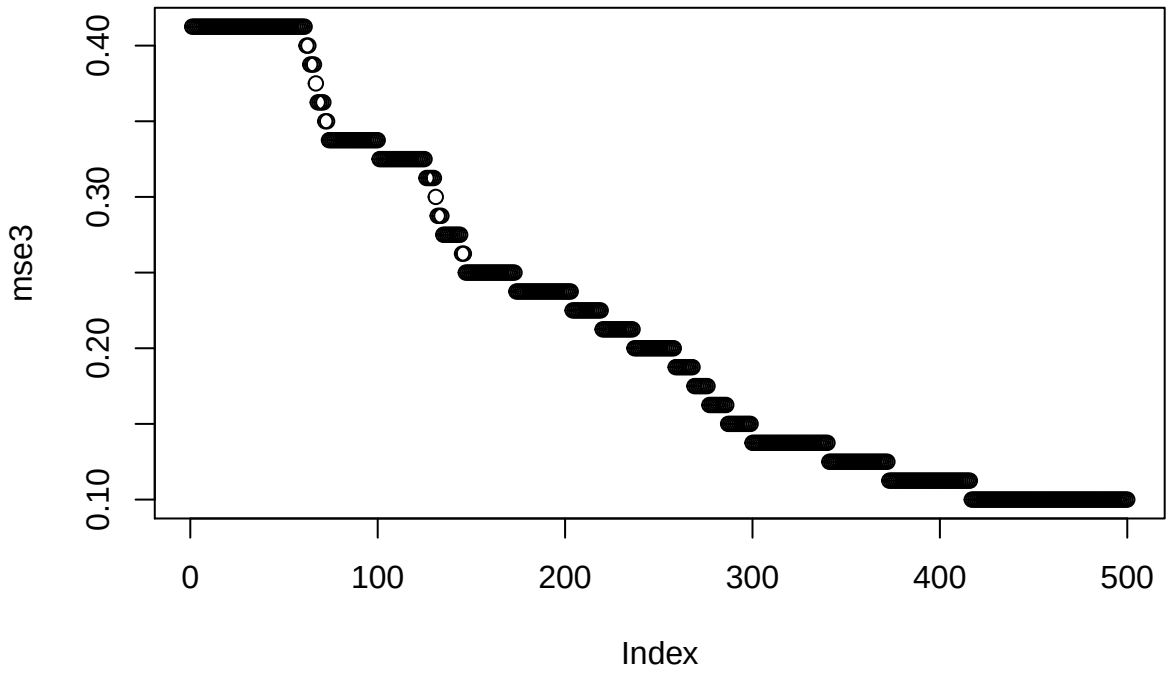
```
summary(boosting3)
```




```
##                    var    rel.inf
## Price            Price 30.7075820
## ShelveLoc     ShelveLoc 23.4348672
## CompPrice     CompPrice 13.6720242
## Advertising Advertising 13.1490970
## Age                 Age  9.4924069
## Income           Income  7.0694535
## Population   Population  1.2763808
## Education     Education  0.8026839
## Urban             Urban  0.2155784
## US                   US  0.1799260
```

```
mse3 <- c()
for(i in 1:500){
  yhat <- predict(boosting3, newdata = carseats[-train, ], n.trees = (10 * i))
  yhat <- (yhat > thresh)
  mse3[i] <- mean(yhat != carseats$High[-train])
}


plot(mse3)
```

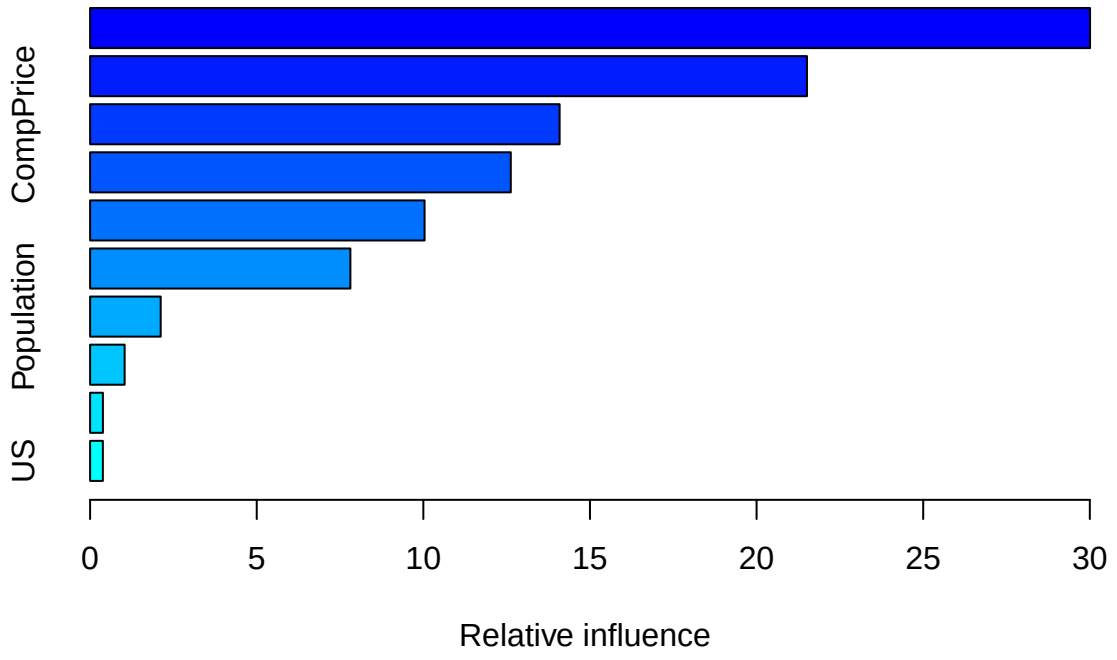



```
#4

boosting4 <- gbm(High~.-Sales, data = carseats[train, ], distribution = "bernoulli",
                 n.trees = 5000, interaction.depth = 4)
boosting4
```

```
## gbm(formula = High ~ . - Sales, distribution = "bernoulli", data = carseats[train,
##     ], n.trees = 5000, interaction.depth = 4)
## A gradient boosted model with bernoulli loss function.
## 5000 iterations were performed.
## There were 10 predictors of which 10 had non-zero influence.
```

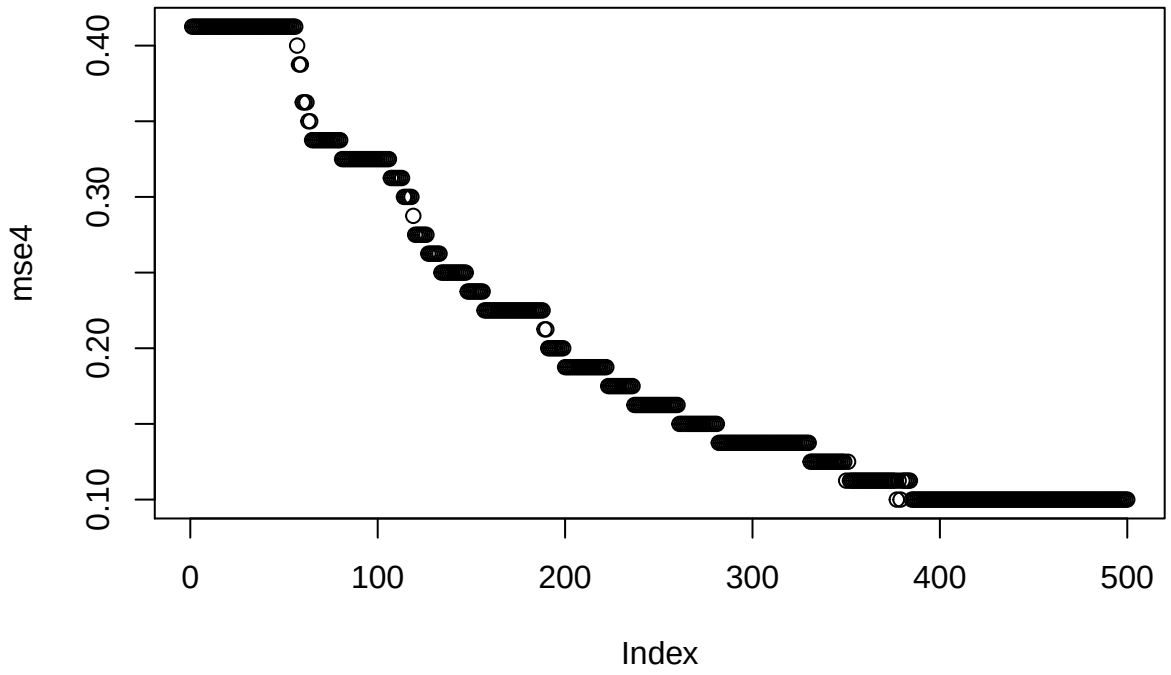
```
summary(boosting4)
```




```
##                     var    rel.inf
## Price             Price 30.0071879
## ShelveLoc     ShelveLoc 21.5113706
## CompPrice     CompPrice 14.0894052
## Advertising Advertising 12.6240706
## Age                 Age 10.0364719
## Income           Income  7.8108277
## Population   Population  2.1178988
## Education     Education  1.0363377
## Urban             Urban  0.3837958
## US                   US  0.3826339
```

```
mse4 <- c()
for(i in 1:500){
  yhat <- predict(boosting4, newdata = carseats[-train, ], n.trees = (10 * i))
  yhat <- (yhat > thresh)
  mse4[i] <- mean(yhat != carseats$High[-train])
}


plot(mse4)
```



ShelveLoc and Price are the top two important varaibles. . . .