Homework 10 - Boosting Methods

Shiloh Bradley

7/1/2020

```r
source("normalize.R")
source("RMSE.R")

library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```r
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```r
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 3.6.2
```

Charles Book Club data

```r
df <- read.csv("Charles_BookClub.csv")
dim(df) ## 2000   18
```

```
## [1] 2000   18
```

```r
names(df)
```

```
##  [1] "Seq."       "ID."        "Gender"     "M"          "R"         
##  [6] "F"          "FirstPurch" "ChildBks"   "YouthBks"   "CookBks"   
## [11] "DoltYBks"   "RefBks"     "ArtBks"     "GeogBks"    "ItalCook"  
## [16] "ItalHAtlas" "ItalArt"    "Florence"
```

```r
head(df)
```

```
##   Seq. ID. Gender   M  R F FirstPurch ChildBks YouthBks CookBks DoltYBks
## 1    1   2      0 138 28 3         40        0        1       0        1
## 2    2  30      1 240 14 1         14        1        0       0        0
## 3    3  59      1  97  6 2         10        0        0       0        0
## 4    4  89      1 348  2 7         38        1        1       1        0
## 5    5  96      0 239 20 2         28        0        0       1        0
## 6    6 120      1 253 10 4         20        1        0       0        0
##   RefBks ArtBks GeogBks ItalCook ItalHAtlas ItalArt Florence
## 1      0      0       1        0          0       0        0
## 2      0      0       0        0          0       0        0
## 3      0      0       0        0          0       0        0
## 4      1      0       1        0          0       0        0
## 5      0      0       1        0          0       0        0
## 6      0      1       0        0          0       0        1
```

```r
summary(df)
```

```
##       Seq.             ID.           Gender            M
##  Min.   :   1.0   Min.   :    2   Min.   :0.0000   Min.   : 15.0
##  1st Qu.: 500.8   1st Qu.:12699   1st Qu.:0.0000   1st Qu.:126.8
##  Median :1000.5   Median :24201   Median :1.0000   Median :207.0
##  Mean   :1000.5   Mean   :24753   Mean   :0.7085   Mean   :206.8
##  3rd Qu.:1500.2   3rd Qu.:37300   3rd Qu.:1.0000   3rd Qu.:281.2
##  Max.   :2000.0   Max.   :49962   Max.   :1.0000   Max.   :477.0
##        R               F             FirstPurch       ChildBks
##  Min.   : 2.00   Min.   : 1.000   Min.   : 2.00   Min.   :0.000
##  1st Qu.: 8.00   1st Qu.: 1.000   1st Qu.:14.00   1st Qu.:0.000
##  Median :12.00   Median : 2.000   Median :22.00   Median :0.000
##  Mean   :13.52   Mean   : 4.005   Mean   :27.42   Mean   :0.711
##  3rd Qu.:16.00   3rd Qu.: 6.000   3rd Qu.:38.00   3rd Qu.:1.000
##  Max.   :36.00   Max.   :12.000   Max.   :99.00   Max.   :6.000
##     YouthBks        CookBks          DoltYBks         RefBks
##  Min.   :0.000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000   Median :0.000   Median :0.0000
##  Mean   :0.314   Mean   :0.7385   Mean   :0.391   Mean   :0.2705
##  3rd Qu.:0.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :5.000   Max.   :8.0000   Max.   :5.000   Max.   :4.0000
##      ArtBks          GeogBks           ItalCook        ItalHAtlas
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :0.3145   Mean   :0.4115   Mean   :0.1285   Mean   :0.0395
##  3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :5.0000   Max.   :5.0000   Max.   :2.0000   Max.   :2.0000
##     ItalArt         Florence
##  Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.000   Median :0.0000
##  Mean   :0.052   Mean   :0.1085
##  3rd Qu.:0.000   3rd Qu.:0.0000
##  Max.   :2.000   Max.   :1.0000
```

```
predictors.cat <- c("Gender", "ChildBks", "YouthBks", "CookBks", "DoltYBks", "RefBks", "ArtBks", "GeogB
predictors.con <- c("Seq.", "ID.", "M", "R", "F", "FirstPurch")
df.cat <- df[predictors.cat]
df.con <- df[predictors.con]
```

```
df.Z <- apply(df[predictors.con], 2, normalize)
summary(df.Z)
```

```
##       Seq.           ID.                M                R
##  Min.   :0.00   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.25   1st Qu.:0.2541   1st Qu.:0.2419   1st Qu.:0.1765
##  Median :0.50   Median :0.4844   Median :0.4156   Median :0.2941
##  Mean   :0.50   Mean   :0.4954   Mean   :0.4151   Mean   :0.3388
##  3rd Qu.:0.75   3rd Qu.:0.7465   3rd Qu.:0.5763   3rd Qu.:0.4118
##  Max.   :1.00   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##        F             FirstPurch
##  Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.1237
##  Median :0.09091   Median :0.2062
##  Mean   :0.27323   Mean   :0.2620
##  3rd Qu.:0.45455   3rd Qu.:0.3711
##  Max.   :1.00000   Max.   :1.0000
```

```
df.cat <- dummy.data.frame(df.cat, sep = ".")
head(df.cat)
```

```
##   Gender ChildBks YouthBks CookBks DoltYBks RefBks ArtBks GeogBks ItalCook
## 1      0        0        1       0        1      0      0       1        0
## 2      1        1        0       0        0      0      0       0        0
## 3      1        0        0       0        0      0      0       0        0
## 4      1        1        1       1        0      1      0       1        0
## 5      0        0        0       1        0      0      0       1        0
## 6      1        1        0       0        0      0      1       0        0
##   ItalHAtlas ItalArt Florence
## 1          0       0        0
## 2          0       0        0
## 3          0       0        0
## 4          0       0        0
## 5          0       0        0
## 6          0       0        1
```

```
df <- cbind.data.frame(df$Florence, df.cat, df.Z)
colnames(df)[1] <- "Florence"
head(df)
```

```
##   Florence Gender ChildBks YouthBks CookBks DoltYBks RefBks ArtBks GeogBks
## 1        0      0        0        1       0        1      0      0       1
## 2        0      1        1        0       0        0      0      0       0
## 3        0      1        0        0       0        0      0      0       0
## 4        0      1        1        1       1        0      1      0       1
## 5        0      0        0        0       1        0      0      0       1
## 6        1      1        1        0       0        0      0      1       0
##   ItalCook ItalHAtlas ItalArt Florence         Seq.          ID.         M
## 1        0          0       0        0 0.0000000000 0.0000000000 0.2662338
## 2        0          0       0        0 0.0005002501 0.0005604484 0.4870130
## 3        0          0       0        0 0.0010005003 0.0011409127 0.1774892
## 4        0          0       0        0 0.0015007504 0.0017413931 0.7207792
## 5        0          0       0        0 0.0020010005 0.0018815052 0.4848485
## 6        0          0       0        1 0.0025012506 0.0023618895 0.5151515
##           R          F FirstPurch
## 1 0.7647059 0.18181818 0.39175258
## 2 0.3529412 0.00000000 0.12371134
## 3 0.1176471 0.09090909 0.08247423
## 4 0.0000000 0.54545455 0.37113402
## 5 0.5294118 0.09090909 0.26804124
## 6 0.2352941 0.27272727 0.18556701
```

```
M <- trunc(.25 * nrow(df))

# to be able to replicate the results, set initial seed for random
# number generator
set.seed(1797317)
holdout <- sample(1:nrow(df), M, replace = F)
df.train <- df[-holdout, ]
df.test <- df[holdout, ]
```

```r
dim(df.train)
```

```
## [1] 1500   19
```

```r
dim(df.test)
```

```
## [1] 500  19
```

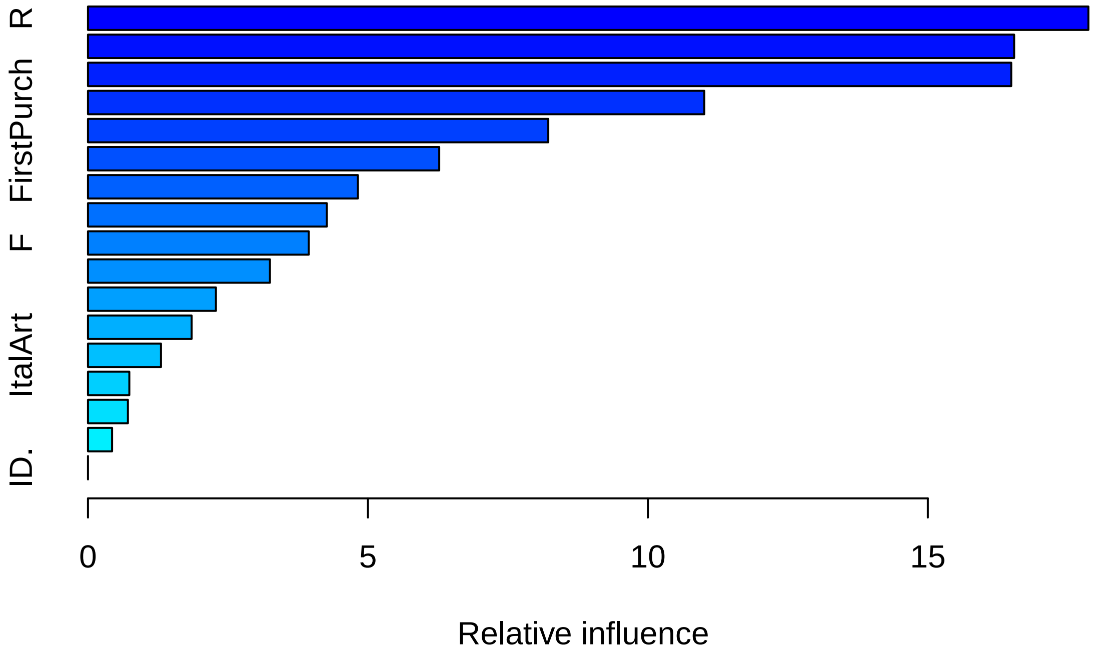
```r
features0 <- setdiff(names(df), c("Florence"))
Formula0 <- formula(paste("Florence ~ ",
                          paste(features0, collapse = " + ")))
Formula0
```

```
## Florence ~ Gender + ChildBks + YouthBks + CookBks + DoltYBks +
##     RefBks + ArtBks + GeogBks + ItalCook + ItalHAtlas + ItalArt +
##     Seq. + ID. + M + R + F + FirstPurch
```

```r
gbm1 <- gbm(
  Formula0,
  data = df.train,
  distribution = "gaussian",
  n.trees = 10000,
  shrinkage = 0.001,
  interaction.depth = 4,
  n.cores = NULL, # will use all cores by default
  verbose = FALSE
  )
# print results
print(gbm1)
```

```
## gbm(formula = Formula0, distribution = "gaussian", data = df.train,
##     n.trees = 10000, interaction.depth = 4, shrinkage = 0.001,
##     verbose = FALSE, n.cores = NULL)
## A gradient boosted model with gaussian loss function.
## 10000 iterations were performed.
## There were 17 predictors of which 16 had non-zero influence.
```

```r
smreGB1 <- summary(gbm1)
```



```
str(smreGB1)
```

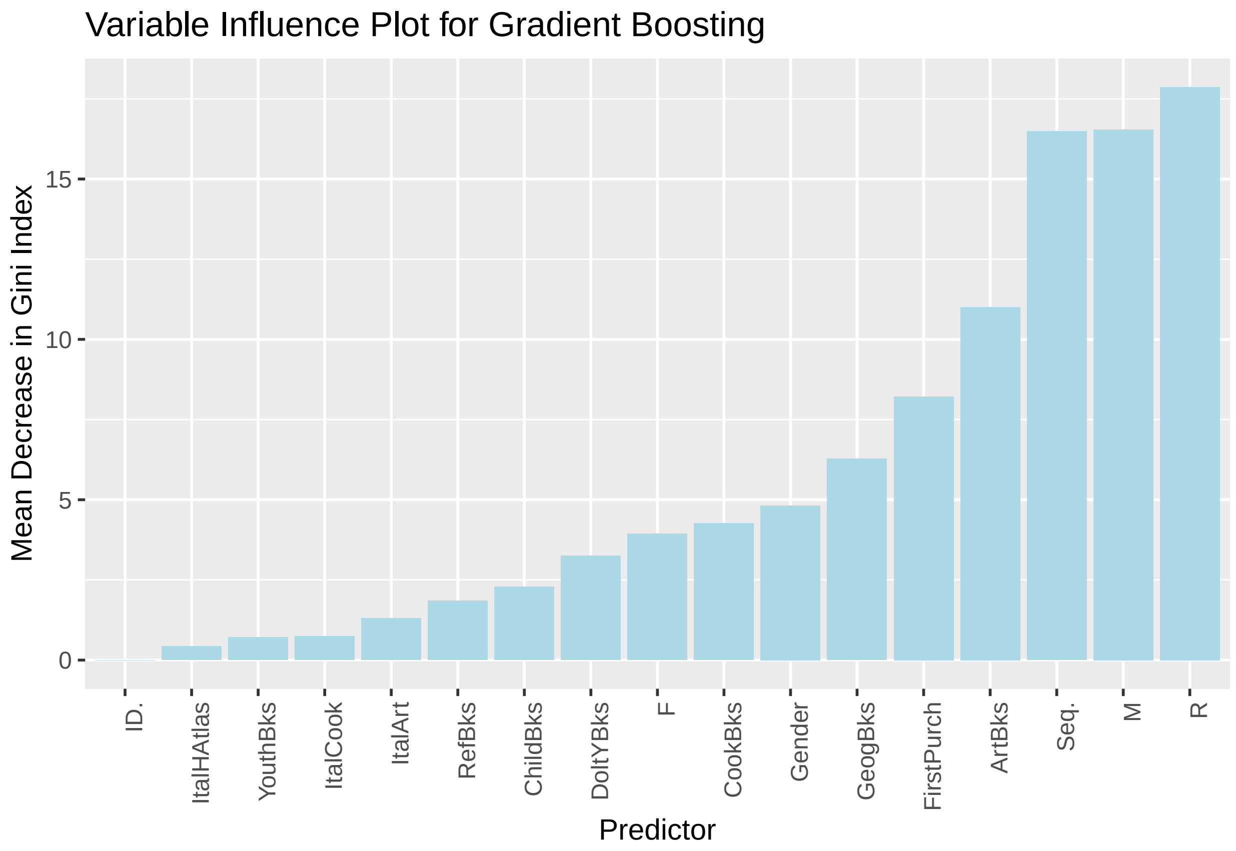
```
## 'data.frame':    17 obs. of  2 variables:
##  $ var    : Factor w/ 17 levels "ArtBks","ChildBks",..: 14 13 16 1 6 8 7 3 5 4 ...
##  $ rel.inf: num  17.87 16.54 16.49 11.01 8.22 ...
```

```
names(smreGB1)
```

```
## [1] "var"     "rel.inf"
```

```
inf.sort <- smreGB1[order(smreGB1[ ,"rel.inf"]), , drop = FALSE]
#write.csv(VIrf1.sort,"VIrf1 120118.csv")
inf.sort$X <- rownames(inf.sort)
inf.sort$X <- factor(inf.sort$X, levels = inf.sort$X)

# Influence Plot in ggplot2
ggplot(inf.sort, aes(x = X, y = rel.inf)) +
    geom_bar(stat = "identity", position = "dodge", fill = "lightblue") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    ylab("Mean Decrease in Gini Index") +
    xlab("Predictor") +
    ggtitle("Variable Influence Plot for Gradient Boosting")
```



```
Y.train <- df.train$Florence
Y.test <- df.test$Florence
Yhat.train_gbm <- gbm1$fit
Yhat.test_gbm <- predict(gbm1, n.trees = gbm1$n.trees, df.test)
RMSE.train_gbm <- RMSE(Y.train, Yhat.train_gbm)
RMSE.test_gbm <- RMSE(Y.test, Yhat.test_gbm)
df.RMSE_gbm <- rbind.data.frame(RMSE.train_gbm, RMSE.test_gbm)
colnames(df.RMSE_gbm) <- c("gbm.R_Square", "gbm.RMSE")
rownames(df.RMSE_gbm) <- c("train", "test")
df.RMSE_gbm
```

```
##       gbm.R_Square  gbm.RMSE
## train    0.3464004 0.2515933
## test     0.1195567 0.3120350
```

```
train.y <- df.train$Florence
test.y <- df.test$Florence

E2.train <- as.matrix(df.train[,-1])
E2.test <- as.matrix(df.test[,-1])

dTrain <- xgb.DMatrix(data = E2.train, label= train.y)  # this specifies response is Train.Y
dTest <- xgb.DMatrix(data = E2.test, label= test.y)  # this specifies response is Test.Y
```

```
set.seed(311317)
searchGridSubCol <- expand.grid(subsample = c(0.5, 0.6),
                                colsample_bytree = c(0.5, 0.6),
                                max_depth = c(3, 4),
                                min_child = seq(1),
                                eta = c(0.1)
)
```

```
set.seed(11317)
searchGridSubCol <- expand.grid(subsample = c(0.5, 0.6),
                                colsample_bytree = c(0.5, 0.6),
                                max_depth = c(3, 4),
                                min_child = seq(1),
                                eta = c(0.1)
)
ntrees <- 50

system.time(
rmseErrorsHyperparameters <- apply(searchGridSubCol, 1, function(parameterList) {

  #Extract Parameters to test
  currentSubsampleRate <- parameterList[["subsample"]]
  currentColsampleRate <- parameterList[["colsample_bytree"]]
  currentDepth <- parameterList[["max_depth"]]
  currentEta <- parameterList[["eta"]]
  currentMinChild <- parameterList[["min_child"]]
  xgboostModelCV <- xgb.cv(data =  dTrain, nrounds = ntrees, nfold = 5, showsd = TRUE,
                       metrics = "rmse", verbose = TRUE, "eval_metric" = "rmse",
                     "objective" = "reg:linear", "max.depth" = currentDepth, "eta" = currentEta,
                     "subsample" = currentSubsampleRate, "colsample_bytree" = currentColsampleRate,
                     print_every_n = 10, "min_child_weight" = currentMinChild, booster = "gbtree",
                     early_stopping_rounds = 10)

  xvalidationScores <- as.data.frame(xgboostModelCV$evaluation_log)
  rmse <- tail(xvalidationScores$test_rmse_mean, 1)
  trmse <- tail(xvalidationScores$train_rmse_mean,1)
  output <- return(c(rmse, trmse, currentSubsampleRate, currentColsampleRate, currentDepth, currentEta,
```

```
## [20:40:53] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:53] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:53] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:53] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:53] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.457668+0.009200    test-rmse:0.458224+0.009940
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.214976+0.027893    test-rmse:0.217959+0.026669
## [21] train-rmse:0.113421+0.020565    test-rmse:0.118763+0.019199
## [31] train-rmse:0.070384+0.018329    test-rmse:0.077062+0.016774
## [41] train-rmse:0.042240+0.011256    test-rmse:0.050262+0.011465
## [50] train-rmse:0.028608+0.006937    test-rmse:0.037089+0.008167
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.457164+0.008611    test-rmse:0.457504+0.009132
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.206361+0.007002    test-rmse:0.208357+0.007732
## [21] train-rmse:0.112049+0.016993    test-rmse:0.117523+0.020780
## [31] train-rmse:0.067656+0.009856    test-rmse:0.073911+0.012390
## [41] train-rmse:0.044186+0.006254    test-rmse:0.050580+0.007561
## [50] train-rmse:0.030846+0.003917    test-rmse:0.037673+0.006021
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:54] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.456939+0.008295    test-rmse:0.457631+0.009154
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.211672+0.012538    test-rmse:0.216540+0.015247
## [21] train-rmse:0.107885+0.015244    test-rmse:0.114292+0.017057
## [31] train-rmse:0.056107+0.017462    test-rmse:0.062671+0.017563
## [41] train-rmse:0.035040+0.012360    test-rmse:0.042220+0.014257
## [50] train-rmse:0.024301+0.007719    test-rmse:0.031855+0.009868
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.457054+0.008473    test-rmse:0.457978+0.009671
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.198838+0.024129    test-rmse:0.201789+0.024897
## [21] train-rmse:0.091876+0.017340    test-rmse:0.096169+0.018203
## [31] train-rmse:0.054903+0.013655    test-rmse:0.060392+0.014790
## [41] train-rmse:0.032798+0.008933    test-rmse:0.038892+0.010229
## [50] train-rmse:0.023030+0.005907    test-rmse:0.029376+0.007051
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.464120+0.007002    test-rmse:0.466173+0.008114
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.204739+0.020006    test-rmse:0.209946+0.019558
## [21] train-rmse:0.098434+0.015915    test-rmse:0.108389+0.017121
## [31] train-rmse:0.058710+0.010057    test-rmse:0.071334+0.012824
## [41] train-rmse:0.036691+0.008972    test-rmse:0.049360+0.010121
## [50] train-rmse:0.027050+0.008828    test-rmse:0.040105+0.010038
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:55] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.453312+0.006337    test-rmse:0.453800+0.007317
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.186645+0.004308    test-rmse:0.189352+0.004859
## [21] train-rmse:0.092006+0.003922    test-rmse:0.099048+0.008146
## [31] train-rmse:0.053696+0.006946    test-rmse:0.063068+0.010694
## [41] train-rmse:0.032506+0.004970    test-rmse:0.042429+0.008992
## [50] train-rmse:0.025923+0.004216    test-rmse:0.036313+0.007851
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.463345+0.006614    test-rmse:0.464502+0.007341
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.212747+0.015414    test-rmse:0.221392+0.024445
## [21] train-rmse:0.096711+0.008115    test-rmse:0.107655+0.015015
## [31] train-rmse:0.050435+0.004458    test-rmse:0.062778+0.007800
## [41] train-rmse:0.031958+0.004003    test-rmse:0.045201+0.007058
## [50] train-rmse:0.023277+0.003445    test-rmse:0.037020+0.007288
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:40:56] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:0.450140+0.000004    test-rmse:0.450140+0.000009
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:0.208594+0.017304    test-rmse:0.214763+0.018663
## [21] train-rmse:0.105825+0.013901    test-rmse:0.115826+0.018018
## [31] train-rmse:0.053701+0.010575    test-rmse:0.064973+0.012119
## [41] train-rmse:0.034089+0.006564    test-rmse:0.046633+0.008536
## [50] train-rmse:0.023138+0.005092    test-rmse:0.036020+0.007340
```

```
##    user  system elapsed
##   9.416   0.406   3.229
```

```r
output <- as.data.frame(t(rmseErrorsHyperparameters))
varnames <- c("TestRMSE", "TrainRMSE", "SubSampRate", "ColSampRate", "Depth", "eta", "currentMinChild")
names(output) <- varnames
output # ntree = 50
```

```
##    TestRMSE TrainRMSE SubSampRate ColSampRate Depth eta currentMinChild
## 1 0.0370892 0.0286078         0.5         0.5     3 0.1               1
## 2 0.0376726 0.0308464         0.6         0.5     3 0.1               1
## 3 0.0318546 0.0243006         0.5         0.6     3 0.1               1
## 4 0.0293756 0.0230296         0.6         0.6     3 0.1               1
## 5 0.0401052 0.0270496         0.5         0.5     4 0.1               1
## 6 0.0363126 0.0259228         0.6         0.5     4 0.1               1
## 7 0.0370198 0.0232768         0.5         0.6     4 0.1               1
## 8 0.0360196 0.0231384         0.6         0.6     4 0.1               1
```

```r
#Final xgboost model
set.seed(11371)
ntree <- 50
xgbF <- xgboost(data = dTrain, # the data
                 nround = 100, # max number of boosting iterations
                 SubSampRate = 0.6,
                 ColSampRate = 0.6,
                 Depth = 4,
                 eta = 0.1,
                 currentMinChild = 1,
                 objective = "reg:linear")  # the objective function
```

```
## [20:40:57] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecate
## [20:40:57] WARNING: amalgamation/../src/learner.cc:480:
## Parameters: { ColSampRate, Depth, SubSampRate, currentMinChild } might not be used.
##
##   This may not be accurate due to some parameters are only used in language bindings but
##   passed down to XGBoost core.  Or some parameters are not used but slip through this
##   verification. Please open an issue if you find above cases.
##
##
## [1]  train-rmse:0.450066
## [2]  train-rmse:0.405119
## [3]  train-rmse:0.364662
## [4]  train-rmse:0.328244
## [5]  train-rmse:0.295464
## [6]  train-rmse:0.265956
## [7]  train-rmse:0.239396
## [8]  train-rmse:0.215488
## [9]  train-rmse:0.193968
## [10] train-rmse:0.174597
## [11] train-rmse:0.157161
## [12] train-rmse:0.141466
## [13] train-rmse:0.127338
## [14] train-rmse:0.114621
## [15] train-rmse:0.103174
## [16] train-rmse:0.092871
## [17] train-rmse:0.083596
## [18] train-rmse:0.075248
## [19] train-rmse:0.067733
## [20] train-rmse:0.060969
## [21] train-rmse:0.054880
## [22] train-rmse:0.049400
## [23] train-rmse:0.044466
## [24] train-rmse:0.040026
## [25] train-rmse:0.036028
## [26] train-rmse:0.032430
## [27] train-rmse:0.029192
## [28] train-rmse:0.026276
## [29] train-rmse:0.023652
## [30] train-rmse:0.021290
## [31] train-rmse:0.019164
## [32] train-rmse:0.017250
## [33] train-rmse:0.015528
## [34] train-rmse:0.013977
## [35] train-rmse:0.012581
## [36] train-rmse:0.011325
## [37] train-rmse:0.010194
## [38] train-rmse:0.009176
## [39] train-rmse:0.008259
## [40] train-rmse:0.007435
## [41] train-rmse:0.006692
## [42] train-rmse:0.006024
## [43] train-rmse:0.005422
## [44] train-rmse:0.004881
## [45] train-rmse:0.004393
## [46] train-rmse:0.003955
## [47] train-rmse:0.003560
## [48] train-rmse:0.003204
## [49] train-rmse:0.002884
## [50] train-rmse:0.002596
## [51] train-rmse:0.002337
## [52] train-rmse:0.002104
## [53] train-rmse:0.001894
## [54] train-rmse:0.001704
## [55] train-rmse:0.001534
## [56] train-rmse:0.001381
## [57] train-rmse:0.001243
## [58] train-rmse:0.001119
## [59] train-rmse:0.001007
## [60] train-rmse:0.000907
## [61] train-rmse:0.000816
## [62] train-rmse:0.000735
## [63] train-rmse:0.000661
## [64] train-rmse:0.000595
## [65] train-rmse:0.000536
## [66] train-rmse:0.000482
## [67] train-rmse:0.000434
## [68] train-rmse:0.000391
## [69] train-rmse:0.000352
## [70] train-rmse:0.000317
## [71] train-rmse:0.000285
## [72] train-rmse:0.000257
## [73] train-rmse:0.000231
## [74] train-rmse:0.000208
## [75] train-rmse:0.000187
## [76] train-rmse:0.000168
## [77] train-rmse:0.000152
## [78] train-rmse:0.000136
## [79] train-rmse:0.000123
## [80] train-rmse:0.000111
## [81] train-rmse:0.000100
## [82] train-rmse:0.000090
## [83] train-rmse:0.000081
## [84] train-rmse:0.000073
## [85] train-rmse:0.000065
## [86] train-rmse:0.000059
## [87] train-rmse:0.000053
## [88] train-rmse:0.000048
## [89] train-rmse:0.000043
## [90] train-rmse:0.000039
## [91] train-rmse:0.000036
## [92] train-rmse:0.000034
## [93] train-rmse:0.000033
## [94] train-rmse:0.000031
## [95] train-rmse:0.000030
## [96] train-rmse:0.000029
## [97] train-rmse:0.000028
## [98] train-rmse:0.000027
## [99] train-rmse:0.000027
## [100]    train-rmse:0.000026
```

```r
pred.train_xgb <- predict(xgbF, dTrain)
pred.test_xgb <- predict(xgbF, dTest)
#RMSE <- function(Y,Yhat)
RMSR.train_xgb <- RMSE(train.y,pred.train_xgb)
RMSR.test_xgb <- RMSE(test.y,pred.test_xgb)
RMSR.train_xgb
```

```
##    R_Square        RMSE
## 1.00000e+00 2.62138e-05
```

```r
RMSR.test_xgb
```

```
##     R_Square         RMSE
## 1.000000e+00 2.760096e-05
```

Boston Housing Data

```r
df <- read.csv("Boston Housing.csv")
dim(df) ## 506  15
```

```
## [1] 506  15
```

```r
names(df)
```

```
##  [1] "X"       "crim"    "zn"      "indus"   "chas"    "nox"     "rm"
##  [8] "age"     "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"
## [15] "medv"
```

```r
head(df)
```

```
##   X    crim zn indus chas   nox    rm  age    dis rad tax ptratio  black
## 1 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
## 2 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
## 3 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
## 4 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
## 5 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
## 6 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
##   lstat medv
## 1  4.98 24.0
## 2  9.14 21.6
## 3  4.03 34.7
## 4  2.94 33.4
## 5  5.33 36.2
## 6  5.21 28.7
```

```r
summary(df)
```

```
##        X              crim                zn             indus      
##  Min.   :  1.0   Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46  
##  1st Qu.:127.2   1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19  
##  Median :253.5   Median : 0.25651   Median :  0.00   Median : 9.69  
##  Mean   :253.5   Mean   : 3.61352   Mean   : 11.36   Mean   :11.14  
##  3rd Qu.:379.8   3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10  
##  Max.   :506.0   Max.   :88.97620   Max.   :100.00   Max.   :27.74  
##       chas              nox               rm             age        
##  Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90  
##  1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02  
##  Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50  
##  Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57  
##  3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08  
##  Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00  
##       dis              rad              tax           ptratio     
##  Min.   : 1.130   Min.   : 1.000   Min.   :187.0   Min.   :12.60  
##  1st Qu.: 2.100   1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40  
##  Median : 3.207   Median : 5.000   Median :330.0   Median :19.05  
##  Mean   : 3.795   Mean   : 9.549   Mean   :408.2   Mean   :18.46  
##  3rd Qu.: 5.188   3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20  
##  Max.   :12.127   Max.   :24.000   Max.   :711.0   Max.   :22.00  
##      black            lstat            medv      
##  Min.   :  0.32   Min.   : 1.73   Min.   : 5.00  
##  1st Qu.:375.38   1st Qu.: 6.95   1st Qu.:17.02  
##  Median :391.44   Median :11.36   Median :21.20  
##  Mean   :356.67   Mean   :12.65   Mean   :22.53  
##  3rd Qu.:396.23   3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :396.90   Max.   :37.97   Max.   :50.00  
```

```r
# predictors.cat <- c("Gender", "ChildBks", "YouthBks", "CookBks", "DoltYBks", "RefBks", "ArtBks", "Geo
# predictors.con <- c("Seq.", "ID.", "M", "R", "F", "FirstPurch")
# df.cat <- df[predictors.cat]
# df.con <- df[predictors.con]
```

```r
df.Z <- apply(df, 2, normalize)
summary(df.Z)
```

```
##        X             crim                 zn             indus       
##  Min.   :0.00   Min.   :0.0000000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.25   1st Qu.:0.0008511   1st Qu.:0.0000   1st Qu.:0.1734  
##  Median :0.50   Median :0.0028121   Median :0.0000   Median :0.3383  
##  Mean   :0.50   Mean   :0.0405441   Mean   :0.1136   Mean   :0.3914  
##  3rd Qu.:0.75   3rd Qu.:0.0412585   3rd Qu.:0.1250   3rd Qu.:0.6466  
##  Max.   :1.00   Max.   :1.0000000   Max.   :1.0000   Max.   :1.0000  
##       chas              nox               rm              age        
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.00000   1st Qu.:0.1317   1st Qu.:0.4454   1st Qu.:0.4338  
##  Median :0.00000   Median :0.3148   Median :0.5073   Median :0.7683  
##  Mean   :0.06917   Mean   :0.3492   Mean   :0.5219   Mean   :0.6764  
##  3rd Qu.:0.00000   3rd Qu.:0.4918   3rd Qu.:0.5868   3rd Qu.:0.9390  
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000  
##       dis               rad              tax            ptratio
##  Min.   :0.00000  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
##  1st Qu.:0.08826  1st Qu.:0.1304  1st Qu.:0.1756  1st Qu.:0.5106
##  Median :0.18895  Median :0.1739  Median :0.2729  Median :0.6862
##  Mean   :0.24238  Mean   :0.3717  Mean   :0.4222  Mean   :0.6229
##  3rd Qu.:0.36909  3rd Qu.:1.0000  3rd Qu.:0.9141  3rd Qu.:0.8085
##  Max.   :1.00000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
##      black            lstat             medv
##  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
##  1st Qu.:0.9457  1st Qu.:0.1440  1st Qu.:0.2672
##  Median :0.9862  Median :0.2657  Median :0.3600
##  Mean   :0.8986  Mean   :0.3014  Mean   :0.3896
##  3rd Qu.:0.9983  3rd Qu.:0.4201  3rd Qu.:0.4444
##  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
```

```
# df.cat <- dummy.data.frame(df.cat, sep = ".")
# head(df.cat)
df <- cbind.data.frame(df$medv, df.Z)
colnames(df)[1] <- "medv"
head(df)
```

```
##   medv           X         crim   zn      indus chas       nox        rm
## 1 24.0 0.000000000 0.0000000000 0.18 0.06781525    0 0.3148148 0.5775053
## 2 21.6 0.001980198 0.0002359225 0.00 0.24230205    0 0.1728395 0.5479977
## 3 34.7 0.003960396 0.0002356977 0.00 0.24230205    0 0.1728395 0.6943859
## 4 33.4 0.005940594 0.0002927957 0.00 0.06304985    0 0.1502058 0.6585553
## 5 36.2 0.007920792 0.0007050701 0.00 0.06304985    0 0.1502058 0.6871048
## 6 28.7 0.009900990 0.0002644715 0.00 0.06304985    0 0.1502058 0.5497222
##         age       dis        rad        tax   ptratio     black      lstat
## 1 0.6416066 0.2692031 0.00000000 0.20801527 0.2872340 1.0000000 0.08967991
## 2 0.7826982 0.3489620 0.04347826 0.10496183 0.5531915 1.0000000 0.20447020
## 3 0.5993821 0.3489620 0.04347826 0.10496183 0.5531915 0.9897373 0.06346578
## 4 0.4418126 0.4485446 0.08695652 0.06679389 0.6489362 0.9942761 0.03338852
## 5 0.5283213 0.4485446 0.08695652 0.06679389 0.6489362 1.0000000 0.09933775
## 6 0.5746653 0.4485446 0.08695652 0.06679389 0.6489362 0.9929901 0.09602649
##        medv
## 1 0.4222222
## 2 0.3688889
## 3 0.6600000
## 4 0.6311111
## 5 0.6933333
## 6 0.5266667
```

```
M <- trunc(.25 * nrow(df))

# to be able to replicate the results, set initial seed for random
# number generator
set.seed(1797317)
holdout <- sample(1:nrow(df), M, replace = F)
df.train <- df[-holdout, ]
df.test <- df[holdout, ]
dim(df.train)
```

```
## [1] 380  16
```

```r
dim(df.test)
```

```
## [1] 126  16
```

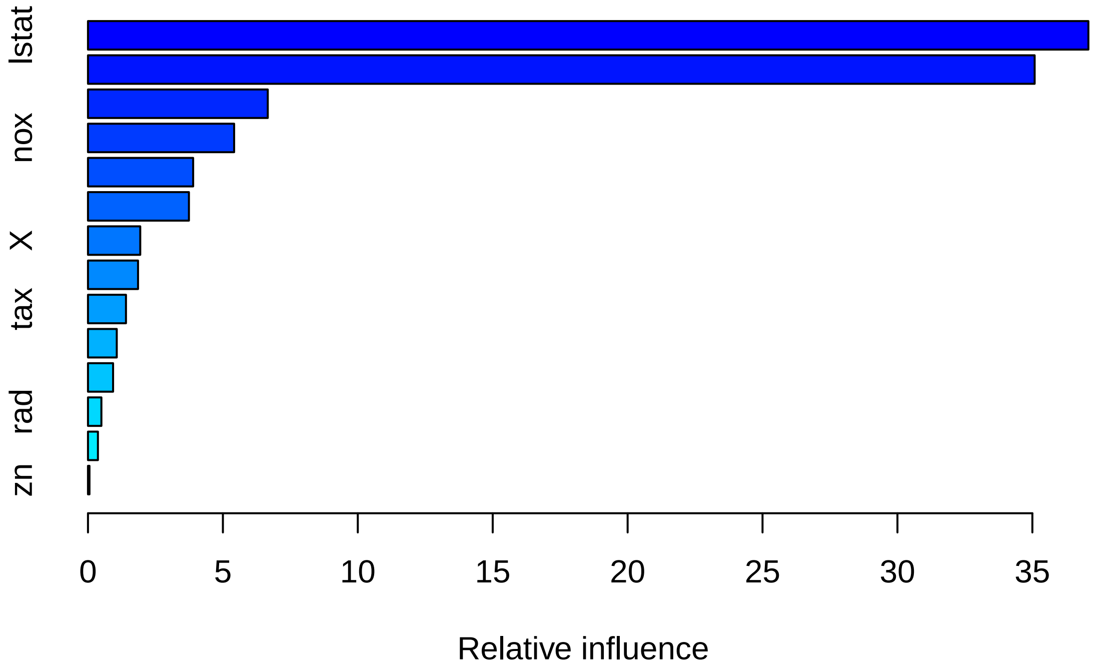
```r
features0 <- setdiff(names(df), c("medv"))
Formula0 <- formula(paste("medv ~ ",
                          paste(features0, collapse = " + ")))
Formula0
```

```
## medv ~ X + crim + zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + black + lstat
```

```r
gbm1 <- gbm(
  Formula0,
  data = df.train,
  distribution = "gaussian",
  n.trees = 10000,
  shrinkage = 0.001,
  interaction.depth = 4,
  n.cores = NULL, # will use all cores by default
  verbose = FALSE
  )
# print results
print(gbm1)
```

```
## gbm(formula = Formula0, distribution = "gaussian", data = df.train,
##     n.trees = 10000, interaction.depth = 4, shrinkage = 0.001,
##     verbose = FALSE, n.cores = NULL)
## A gradient boosted model with gaussian loss function.
## 10000 iterations were performed.
## There were 14 predictors of which 14 had non-zero influence.
```

```r
smreGB1 <- summary(gbm1)
```



```
str(smreGB1)
```

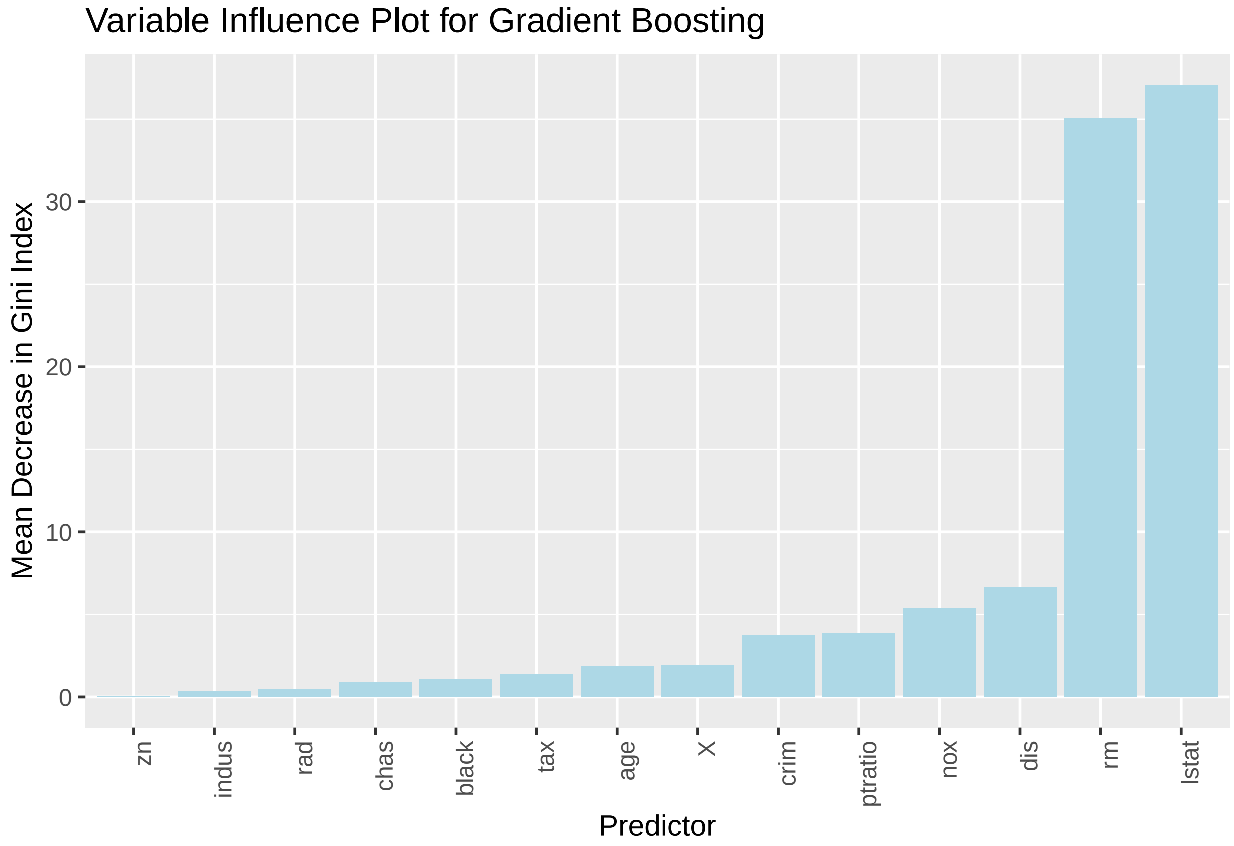
```
## 'data.frame':    14 obs. of  2 variables:
##  $ var    : Factor w/ 14 levels "age","black",..: 7 11 5 8 9 4 13 1 12 2 ...
##  $ rel.inf: num  37.08 35.09 6.66 5.42 3.9 ...
```

```
names(smreGB1)
```

```
## [1] "var"     "rel.inf"
```

```
inf.sort <- smreGB1[order(smreGB1[,"rel.inf"]), , drop = FALSE]
#write.csv(VIrf1.sort,"VIrf1 120118.csv")
inf.sort$X <- rownames(inf.sort)
inf.sort$X <- factor(inf.sort$X, levels = inf.sort$X)

# Influence Plot in ggplot2
ggplot(inf.sort, aes(x = X, y = rel.inf)) +
    geom_bar(stat = "identity", position = "dodge", fill = "lightblue") +
    theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
    ylab("Mean Decrease in Gini Index") +
    xlab("Predictor") +
    ggtitle("Variable Influence Plot for Gradient Boosting")
```



```
Y.train <- df.train$medv
Y.test <- df.test$medv
Yhat.train_gbm <- gbm1$fit
Yhat.test_gbm <- predict(gbm1, n.trees = gbm1$n.trees, df.test)
RMSE.train_gbm <- RMSE(Y.train, Yhat.train_gbm)
RMSE.test_gbm <- RMSE(Y.test, Yhat.test_gbm)
```

```
df.RMSE_gbm <- rbind.data.frame(RMSE.train_gbm, RMSE.test_gbm)
colnames(df.RMSE_gbm) <- c("gbm.R_Square", "gbm.RMSE")
rownames(df.RMSE_gbm) <- c("train", "test")
df.RMSE_gbm
```

```
##       gbm.R_Square gbm.RMSE
## train    0.9497593 2.056837
## test     0.8734814 3.369560
```

```
train.y <- df.train$medv
test.y <- df.test$medv

E2.train <- as.matrix(df.train[,-1])
E2.test <- as.matrix(df.test[,-1])

dTrain <- xgb.DMatrix(data = E2.train, label = train.y)  # this specifies response is Train.Y
dTest <- xgb.DMatrix(data = E2.test, label = test.y)  # this specifies response is Test.Y
```

```
set.seed(311317)
searchGridSubCol <- expand.grid(subsample = c(0.5, 0.6),
                                colsample_bytree = c(0.5, 0.6),
                                max_depth = c(3, 4),
                                min_child = seq(1),
                                eta = c(0.1)
)
```

```
set.seed(11317)
searchGridSubCol <- expand.grid(subsample = c(0.5, 0.6),
                                colsample_bytree = c(0.5, 0.6),
                                max_depth = c(3, 4),
                                min_child = seq(1),
                                eta = c(0.1)
)
ntrees <- 50

system.time(
rmseErrorsHyperparameters <- apply(searchGridSubCol, 1, function(parameterList) {

  #Extract Parameters to test
  currentSubsampleRate <- parameterList[["subsample"]]
  currentColsampleRate <- parameterList[["colsample_bytree"]]
  currentDepth <- parameterList[["max_depth"]]
  currentEta <- parameterList[["eta"]]
  currentMinChild <- parameterList[["min_child"]]
  xgboostModelCV <- xgb.cv(data =  dTrain, nrounds = ntrees, nfold = 5, showsd = TRUE,
                     metrics = "rmse", verbose = TRUE, "eval_metric" = "rmse",
                    "objective" = "reg:linear", "max.depth" = currentDepth, "eta" = currentEta,
                    "subsample" = currentSubsampleRate, "colsample_bytree" = currentColsampleRate,
                    print_every_n = 10, "min_child_weight" = currentMinChild, booster = "gbtree",
                    early_stopping_rounds = 10)

  xvalidationScores <- as.data.frame(xgboostModelCV$evaluation_log)
  rmse <- tail(xvalidationScores$test_rmse_mean, 1)
  trmse <- tail(xvalidationScores$train_rmse_mean,1)
  output <- return(c(rmse, trmse, currentSubsampleRate, currentColsampleRate, currentDepth, currentEta,
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.476705+0.340509   test-rmse:21.412042+1.433430
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:8.347527+0.147813    test-rmse:8.454558+0.964417
## [21] train-rmse:3.560939+0.113613    test-rmse:3.831245+0.609818
## [31] train-rmse:1.838225+0.117063    test-rmse:2.245844+0.413972
## [41] train-rmse:1.257931+0.143195    test-rmse:1.783183+0.340336
## [50] train-rmse:1.006169+0.120179    test-rmse:1.581589+0.290168
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.456433+0.405141   test-rmse:21.397811+1.760326
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:8.185130+0.232512    test-rmse:8.229772+1.019389
## [21] train-rmse:3.390614+0.157097    test-rmse:3.635185+0.714361
## [31] train-rmse:1.638520+0.152178    test-rmse:2.130462+0.605597
## [41] train-rmse:1.060088+0.120369    test-rmse:1.720899+0.537214
## [50] train-rmse:0.848699+0.101838    test-rmse:1.559493+0.480228
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:05] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.471843+0.173360   test-rmse:21.469374+0.879391
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:8.214866+0.100864    test-rmse:8.292992+0.478129
## [21] train-rmse:3.334784+0.056560    test-rmse:3.481744+0.326028
## [31] train-rmse:1.592555+0.078626    test-rmse:1.876432+0.289342
## [41] train-rmse:0.965373+0.072390    test-rmse:1.352992+0.196993
## [50] train-rmse:0.733135+0.058979    test-rmse:1.206067+0.165018
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.481620+0.227376   test-rmse:21.493145+0.999723
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:8.110035+0.094233    test-rmse:8.154945+0.700597
## [21] train-rmse:3.266418+0.120720    test-rmse:3.418234+0.391139
## [31] train-rmse:1.460589+0.105675    test-rmse:1.741751+0.245429
## [41] train-rmse:0.864529+0.121265    test-rmse:1.288732+0.201524
## [50] train-rmse:0.658562+0.107738    test-rmse:1.138507+0.189163
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:06] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.484145+0.246925   test-rmse:21.521057+1.077766
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:8.427072+0.280574    test-rmse:8.562548+0.750965
## [21] train-rmse:3.543777+0.173016    test-rmse:3.921472+0.710421
## [31] train-rmse:1.791378+0.175558    test-rmse:2.427674+0.652238
## [41] train-rmse:1.118126+0.115241    test-rmse:1.936439+0.572535
## [50] train-rmse:0.858001+0.102087    test-rmse:1.769156+0.558389
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.469484+0.344831   test-rmse:21.534815+1.419617
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:8.265989+0.140444    test-rmse:8.490819+0.869223
## [21] train-rmse:3.421641+0.167984    test-rmse:3.790866+0.604455
## [31] train-rmse:1.619228+0.146058    test-rmse:2.207727+0.597713
## [41] train-rmse:0.989132+0.121099    test-rmse:1.738497+0.523985
## [50] train-rmse:0.751585+0.108847    test-rmse:1.591030+0.481668
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.478899+0.385119   test-rmse:21.361377+1.422541
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
##
## [11] train-rmse:8.299277+0.159398    test-rmse:8.516507+1.090180
## [21] train-rmse:3.423463+0.121773    test-rmse:3.799360+0.758593
## [31] train-rmse:1.591101+0.133062    test-rmse:2.109284+0.536552
## [41] train-rmse:0.954164+0.132552    test-rmse:1.613825+0.389544
## [50] train-rmse:0.706062+0.126092    test-rmse:1.436127+0.324036
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:07] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [1]  train-rmse:21.428621+0.105918   test-rmse:21.380660+0.331643
## Multiple eval metrics are present. Will use test_rmse for early stopping.
## Will train until test_rmse hasn't improved in 10 rounds.
```

```
##
## [11] train-rmse:8.108043+0.093978    test-rmse:8.209189+0.295243
## [21] train-rmse:3.256062+0.108440    test-rmse:3.595106+0.492758
## [31] train-rmse:1.521125+0.112249    test-rmse:2.095480+0.455331
## [41] train-rmse:0.852521+0.098185    test-rmse:1.608848+0.417915
## [50] train-rmse:0.612978+0.090567    test-rmse:1.460429+0.409017
```

```
##    user  system elapsed
##   5.770   0.670   2.884
```

```
output <- as.data.frame(t(rmseErrorsHyperparameters))
varnames <- c("TestRMSE", "TrainRMSE", "SubSampRate", "ColSampRate", "Depth", "eta", "currentMinChild")
names(output) <- varnames
output # ntree = 50
```

```
##   TestRMSE TrainRMSE SubSampRate ColSampRate Depth eta currentMinChild
## 1 1.581589 1.0061686         0.5         0.5     3 0.1               1
## 2 1.559493 0.8486992         0.6         0.5     3 0.1               1
## 3 1.206067 0.7331346         0.5         0.6     3 0.1               1
## 4 1.138507 0.6585620         0.6         0.6     3 0.1               1
## 5 1.769156 0.8580010         0.5         0.5     4 0.1               1
## 6 1.591030 0.7515848         0.6         0.5     4 0.1               1
## 7 1.436127 0.7060620         0.5         0.6     4 0.1               1
## 8 1.460429 0.6129778         0.6         0.6     4 0.1               1
```

```
#Final xgboost model
set.seed(11371)
ntree <- 50
xgbF <- xgboost(data = dTrain, # the data
                 nround = 100, # max number of boosting iterations
                 SubSampRate = 0.6,
                 ColSampRate = 0.6,
                 Depth = 4,
                 eta = 0.1,
                 currentMinChild = 1,
                 objective = "reg:linear")  # the objective function
```

```
## [20:41:08] WARNING: amalgamation/../src/objective/regression_obj.cu:170: reg:linear is now deprecated
## [20:41:08] WARNING: amalgamation/../src/learner.cc:480:
## Parameters: { ColSampRate, Depth, SubSampRate, currentMinChild } might not be used.
##
##   This may not be accurate due to some parameters are only used in language bindings but
##   passed down to XGBoost core.  Or some parameters are not used but slip through this
##   verification. Please open an issue if you find above cases.
##
##
## [1]  train-rmse:21.368555
## [2]  train-rmse:19.290569
## [3]  train-rmse:17.415047
## [4]  train-rmse:15.724111
## [5]  train-rmse:14.196708
## [6]  train-rmse:12.818658
## [7]  train-rmse:11.575500
## [8]  train-rmse:10.453153
## [9]  train-rmse:9.440628
## [10] train-rmse:8.526696
## [11] train-rmse:7.700240
## [12] train-rmse:6.954430
## [13] train-rmse:6.281648
## [14] train-rmse:5.673754
## [15] train-rmse:5.125524
## [16] train-rmse:4.630000
## [17] train-rmse:4.182748
## [18] train-rmse:3.778483
## [19] train-rmse:3.413998
## [20] train-rmse:3.084146
## [21] train-rmse:2.786537
## [22] train-rmse:2.517838
## [23] train-rmse:2.275185
## [24] train-rmse:2.056202
## [25] train-rmse:1.858241
## [26] train-rmse:1.679516
## [27] train-rmse:1.518277
## [28] train-rmse:1.372600
## [29] train-rmse:1.240991
## [30] train-rmse:1.122100
## [31] train-rmse:1.014921
## [32] train-rmse:0.917993
## [33] train-rmse:0.830468
## [34] train-rmse:0.751477
## [35] train-rmse:0.680227
## [36] train-rmse:0.615894
## [37] train-rmse:0.557882
## [38] train-rmse:0.505477
## [39] train-rmse:0.458252
## [40] train-rmse:0.415798
## [41] train-rmse:0.377444
## [42] train-rmse:0.342862
## [43] train-rmse:0.311718
## [44] train-rmse:0.283631
## [45] train-rmse:0.258233
## [46] train-rmse:0.235390
## [47] train-rmse:0.214667
## [48] train-rmse:0.196124
## [49] train-rmse:0.179410
## [50] train-rmse:0.164293
## [51] train-rmse:0.150626
## [52] train-rmse:0.138571
## [53] train-rmse:0.127499
## [54] train-rmse:0.117609
## [55] train-rmse:0.108610
## [56] train-rmse:0.100710
## [57] train-rmse:0.093360
## [58] train-rmse:0.086785
## [59] train-rmse:0.080721
## [60] train-rmse:0.075383
## [61] train-rmse:0.070567
## [62] train-rmse:0.066005
## [63] train-rmse:0.062149
## [64] train-rmse:0.058739
## [65] train-rmse:0.055737
## [66] train-rmse:0.052572
## [67] train-rmse:0.049951
## [68] train-rmse:0.047346
## [69] train-rmse:0.044940
## [70] train-rmse:0.042773
## [71] train-rmse:0.040829
## [72] train-rmse:0.039433
## [73] train-rmse:0.038030
## [74] train-rmse:0.036421
## [75] train-rmse:0.035210
## [76] train-rmse:0.034135
## [77] train-rmse:0.033157
## [78] train-rmse:0.032361
## [79] train-rmse:0.031256
## [80] train-rmse:0.030671
## [81] train-rmse:0.030156
## [82] train-rmse:0.029547
## [83] train-rmse:0.028640
## [84] train-rmse:0.028213
## [85] train-rmse:0.027390
## [86] train-rmse:0.027037
## [87] train-rmse:0.026381
## [88] train-rmse:0.025982
## [89] train-rmse:0.025202
## [90] train-rmse:0.024777
## [91] train-rmse:0.024450
## [92] train-rmse:0.023504
## [93] train-rmse:0.023120
## [94] train-rmse:0.022644
## [95] train-rmse:0.022236
## [96] train-rmse:0.021907
## [97] train-rmse:0.021675
## [98] train-rmse:0.021319
## [99] train-rmse:0.021035
## [100]    train-rmse:0.020854
```

```r
pred.train_xgb <- predict(xgbF, dTrain)
pred.test_xgb <- predict(xgbF, dTest)
#RMSE <- function(Y,Yhat)
RMSR.train_xgb <- RMSE(train.y, pred.train_xgb)
RMSR.test_xgb <- RMSE(test.y, pred.test_xgb)
RMSR.train_xgb
```

```
##   R_Square       RMSE
## 0.99999479 0.02085419
```

```r
RMSR.test_xgb
```

```
##  R_Square      RMSE
## 0.9995147 0.2080776
```