project.R

duter

2021-03-08

##### Dependencies #####  
  
R.Version()

## $platform  
## [1] "x86\_64-w64-mingw32"  
##   
## $arch  
## [1] "x86\_64"  
##   
## $os  
## [1] "mingw32"  
##   
## $system  
## [1] "x86\_64, mingw32"  
##   
## $status  
## [1] ""  
##   
## $major  
## [1] "4"  
##   
## $minor  
## [1] "0.3"  
##   
## $year  
## [1] "2020"  
##   
## $month  
## [1] "10"  
##   
## $day  
## [1] "10"  
##   
## $`svn rev`  
## [1] "79318"  
##   
## $language  
## [1] "R"  
##   
## $version.string  
## [1] "R version 4.0.3 (2020-10-10)"  
##   
## $nickname  
## [1] "Bunny-Wunnies Freak Out"

if (!require(tree))  
 install.packages('tree')

## Loading required package: tree

## Warning: package 'tree' was built under R version 4.0.4

library(tree)  
  
if (!require(randomForest))  
 install.packages("randomForest")

## Loading required package: randomForest

## Warning: package 'randomForest' was built under R version 4.0.4

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

library(randomForest)  
  
if (!require(randomForestExplainer))  
 install.packages("randomForestExplainer")

## Loading required package: randomForestExplainer

## Warning: package 'randomForestExplainer' was built under R version 4.0.4

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(randomForestExplainer)  
  
##### Pre-process Data #####  
  
data <-  
 read.csv('ObesityDataSet.csv',  
 na.strings = '?',  
 stringsAsFactors = T)  
data <- na.omit(data)  
  
# rename columns  
colnames(data)[6] <- 'eats\_high\_calor\_food'  
colnames(data)[7] <- 'eats\_veggies'  
colnames(data)[8] <- 'num\_meals'  
colnames(data)[9] <- 'eats\_snacks'  
colnames(data)[11] <- 'drinks\_water'  
colnames(data)[12] <- 'counts\_calories'  
colnames(data)[13] <- 'exercises\_often'  
colnames(data)[14] <- 'time\_using\_tech'  
colnames(data)[15] <- 'drinks\_alcohol'  
colnames(data)[16] <- 'method\_trans'  
colnames(data)[17] <- 'weight\_category'  
names(data)

## [1] "Gender" "Age"   
## [3] "Height" "Weight"   
## [5] "family\_history\_with\_overweight" "eats\_high\_calor\_food"   
## [7] "eats\_veggies" "num\_meals"   
## [9] "eats\_snacks" "SMOKE"   
## [11] "drinks\_water" "counts\_calories"   
## [13] "exercises\_often" "time\_using\_tech"   
## [15] "drinks\_alcohol" "method\_trans"   
## [17] "weight\_category"

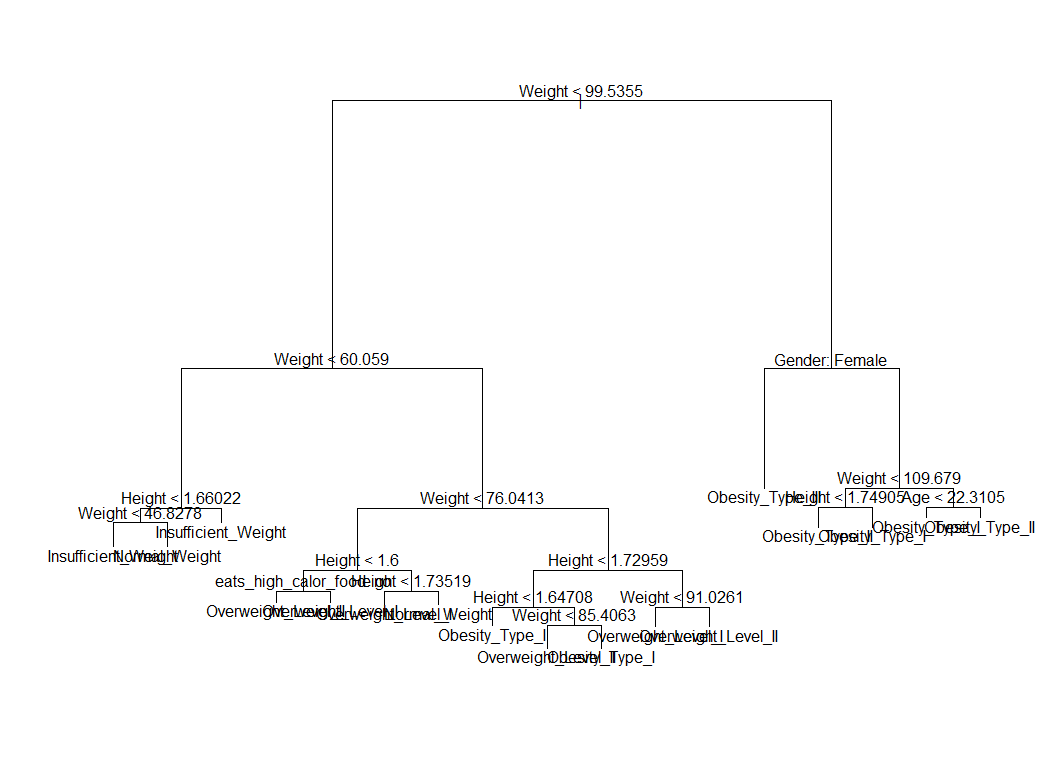
head(data)

## Gender Age Height Weight family\_history\_with\_overweight eats\_high\_calor\_food  
## 1 Female 21 1.62 64.0 yes no  
## 2 Female 21 1.52 56.0 yes no  
## 3 Male 23 1.80 77.0 yes no  
## 4 Male 27 1.80 87.0 no no  
## 5 Male 22 1.78 89.8 no no  
## 6 Male 29 1.62 53.0 no yes  
## eats\_veggies num\_meals eats\_snacks SMOKE drinks\_water counts\_calories  
## 1 2 3 Sometimes no 2 no  
## 2 3 3 Sometimes yes 3 yes  
## 3 2 3 Sometimes no 2 no  
## 4 3 3 Sometimes no 2 no  
## 5 2 1 Sometimes no 2 no  
## 6 2 3 Sometimes no 2 no  
## exercises\_often time\_using\_tech drinks\_alcohol method\_trans  
## 1 0 1 no Public\_Transportation  
## 2 3 0 Sometimes Public\_Transportation  
## 3 2 1 Frequently Public\_Transportation  
## 4 2 0 Frequently Walking  
## 5 0 0 Sometimes Public\_Transportation  
## 6 0 0 Sometimes Automobile  
## weight\_category  
## 1 Normal\_Weight  
## 2 Normal\_Weight  
## 3 Normal\_Weight  
## 4 Overweight\_Level\_I  
## 5 Overweight\_Level\_II  
## 6 Normal\_Weight

set.seed(1)  
  
# split sample and test  
train <- sample(1:nrow(data), nrow(data) / 2)  
test <- data[-train,]  
  
##### Decision Tree #####  
  
# plant a tree  
data.tree <- tree(weight\_category ~ ., data = data)  
  
summary(data.tree)

##   
## Classification tree:  
## tree(formula = weight\_category ~ ., data = data)  
## Variables actually used in tree construction:  
## [1] "Weight" "Height" "eats\_high\_calor\_food"  
## [4] "Gender" "Age"   
## Number of terminal nodes: 17   
## Residual mean deviance: 0.7234 = 1515 / 2094   
## Misclassification error rate: 0.1393 = 294 / 2111

plot(data.tree)  
text(data.tree, pretty = 0)



data.tree <- tree(weight\_category ~ ., data = data, subset = train)  
data.tree.pred = predict(data.tree, test, type = "class")  
  
table(data.tree.pred, test$weight\_category)

##   
## data.tree.pred Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Insufficient\_Weight 138 21 0  
## Normal\_Weight 3 65 0  
## Obesity\_Type\_I 0 0 134  
## Obesity\_Type\_II 0 0 15  
## Obesity\_Type\_III 0 0 0  
## Overweight\_Level\_I 0 44 0  
## Overweight\_Level\_II 0 0 28  
##   
## data.tree.pred Obesity\_Type\_II Obesity\_Type\_III Overweight\_Level\_I  
## Insufficient\_Weight 0 0 0  
## Normal\_Weight 0 0 5  
## Obesity\_Type\_I 8 0 0  
## Obesity\_Type\_II 134 1 0  
## Obesity\_Type\_III 1 176 0  
## Overweight\_Level\_I 0 0 121  
## Overweight\_Level\_II 0 0 17  
##   
## data.tree.pred Overweight\_Level\_II  
## Insufficient\_Weight 0  
## Normal\_Weight 1  
## Obesity\_Type\_I 9  
## Obesity\_Type\_II 0  
## Obesity\_Type\_III 0  
## Overweight\_Level\_I 11  
## Overweight\_Level\_II 124

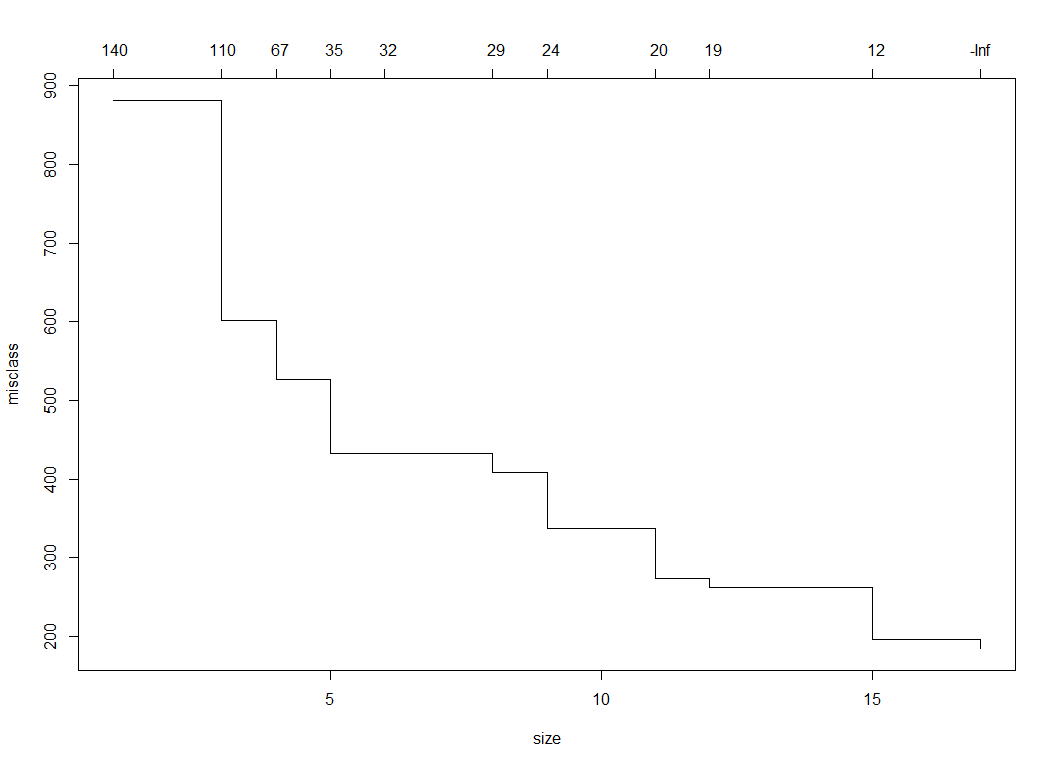
mean(data.tree.pred == test$weight\_category)

## [1] 0.844697

# cross validating different trees   
cv.data <- cv.tree(data.tree, FUN = prune.misclass)  
names(cv.data)

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

plot(cv.data)

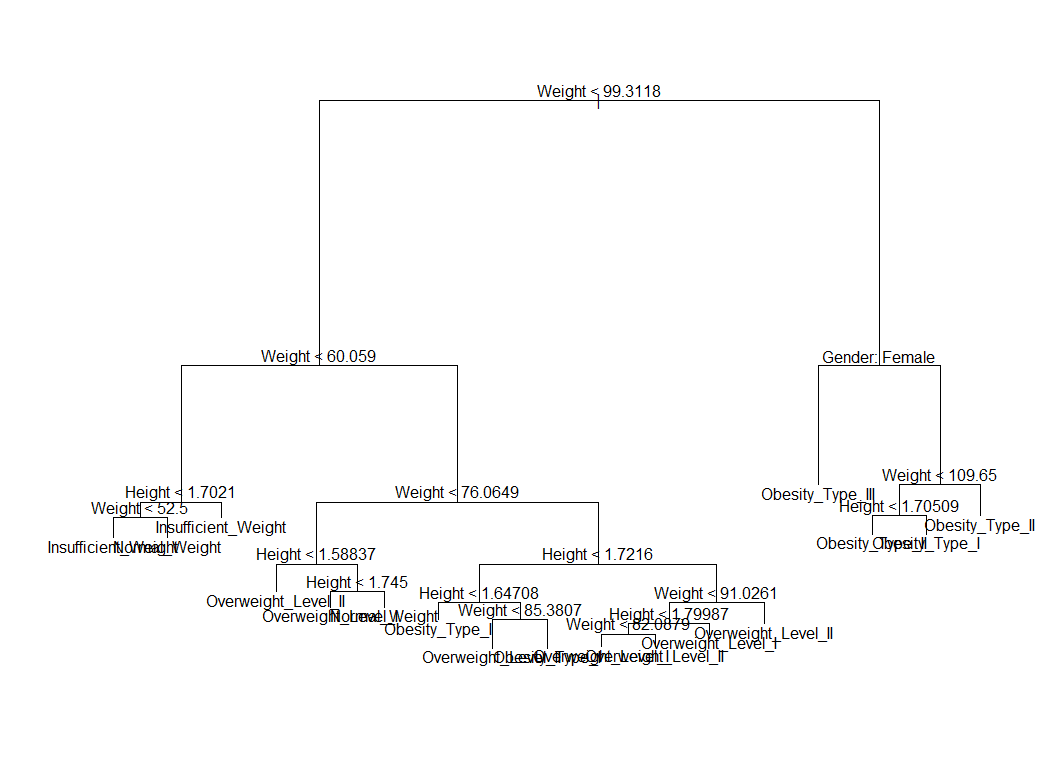


cv.data

## $size  
## [1] 17 15 12 11 9 8 6 5 4 3 1  
##   
## $dev  
## [1] 185 196 262 273 337 408 433 433 527 601 881  
##   
## $k  
## [1] -Inf 11.5 19.0 20.0 23.5 29.0 32.5 35.0 67.0 111.0 141.0  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

# get the best tree  
prune.data <-  
 prune.misclass(data.tree, best = cv.data$size[which.min(cv.data$dev)])

plot(prune.data)  
text(prune.data, pretty = 0)



prune.data.pred <- predict(prune.data, test, type = "class")  
table(prune.data.pred, test$weight\_category)

##   
## prune.data.pred Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Insufficient\_Weight 138 21 0  
## Normal\_Weight 3 65 0  
## Obesity\_Type\_I 0 0 134  
## Obesity\_Type\_II 0 0 15  
## Obesity\_Type\_III 0 0 0  
## Overweight\_Level\_I 0 44 0  
## Overweight\_Level\_II 0 0 28  
##   
## prune.data.pred Obesity\_Type\_II Obesity\_Type\_III Overweight\_Level\_I  
## Insufficient\_Weight 0 0 0  
## Normal\_Weight 0 0 5  
## Obesity\_Type\_I 8 0 0  
## Obesity\_Type\_II 134 1 0  
## Obesity\_Type\_III 1 176 0  
## Overweight\_Level\_I 0 0 121  
## Overweight\_Level\_II 0 0 17  
##   
## prune.data.pred Overweight\_Level\_II  
## Insufficient\_Weight 0  
## Normal\_Weight 1  
## Obesity\_Type\_I 9  
## Obesity\_Type\_II 0  
## Obesity\_Type\_III 0  
## Overweight\_Level\_I 11  
## Overweight\_Level\_II 124

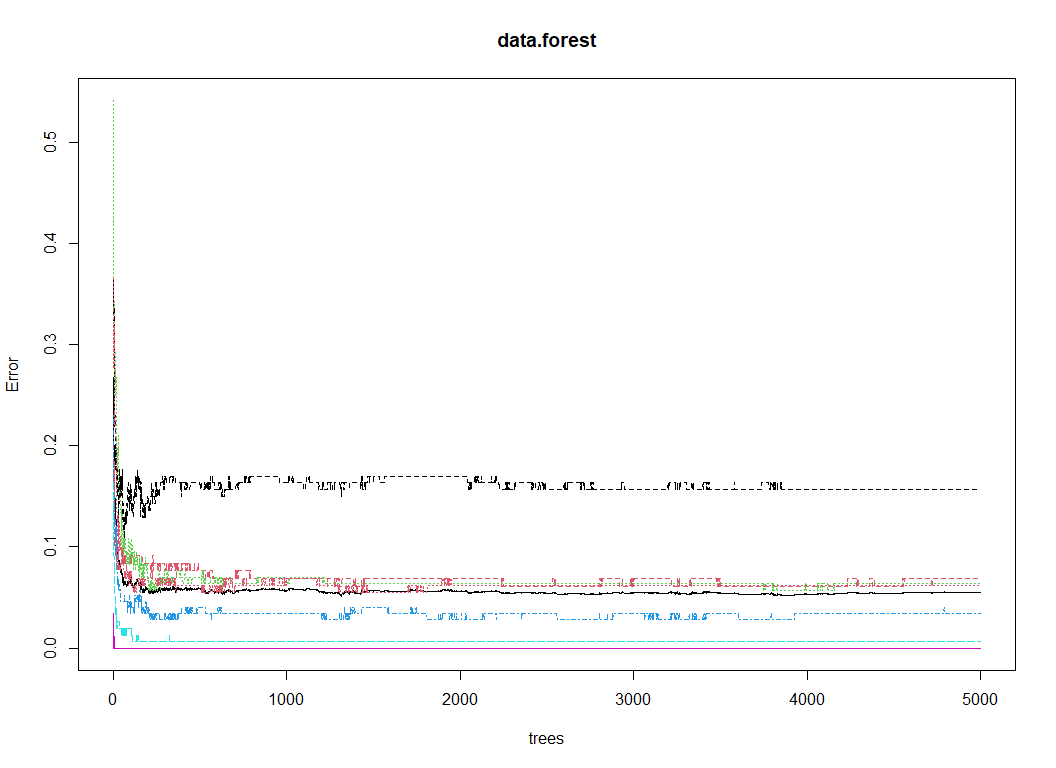
mean(prune.data.pred == test$weight\_category)

## [1] 0.844697

##### Random Forest #####  
  
data.forest <- randomForest(  
 weight\_category ~ .,  
 data = data,  
 subset = train,  
 ntree = 5000,  
 importance = T,  
 localImp = T  
)  
data.forest

##   
## Call:  
## randomForest(formula = weight\_category ~ ., data = data, ntree = 5000, importance = T, localImp = T, subset = train)   
## Type of random forest: classification  
## Number of trees: 5000  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 5.5%  
## Confusion matrix:  
## Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Insufficient\_Weight 122 9 0  
## Normal\_Weight 0 147 0  
## Obesity\_Type\_I 0 1 168  
## Obesity\_Type\_II 0 1 0  
## Obesity\_Type\_III 0 0 0  
## Overweight\_Level\_I 0 20 0  
## Overweight\_Level\_II 0 2 2  
## Obesity\_Type\_II Obesity\_Type\_III Overweight\_Level\_I  
## Insufficient\_Weight 0 0 0  
## Normal\_Weight 0 0 8  
## Obesity\_Type\_I 1 0 2  
## Obesity\_Type\_II 153 0 0  
## Obesity\_Type\_III 0 147 0  
## Overweight\_Level\_I 0 0 124  
## Overweight\_Level\_II 0 0 5  
## Overweight\_Level\_II class.error  
## Insufficient\_Weight 0 0.068702290  
## Normal\_Weight 2 0.063694268  
## Obesity\_Type\_I 2 0.034482759  
## Obesity\_Type\_II 0 0.006493506  
## Obesity\_Type\_III 0 0.000000000  
## Overweight\_Level\_I 3 0.156462585  
## Overweight\_Level\_II 136 0.062068966

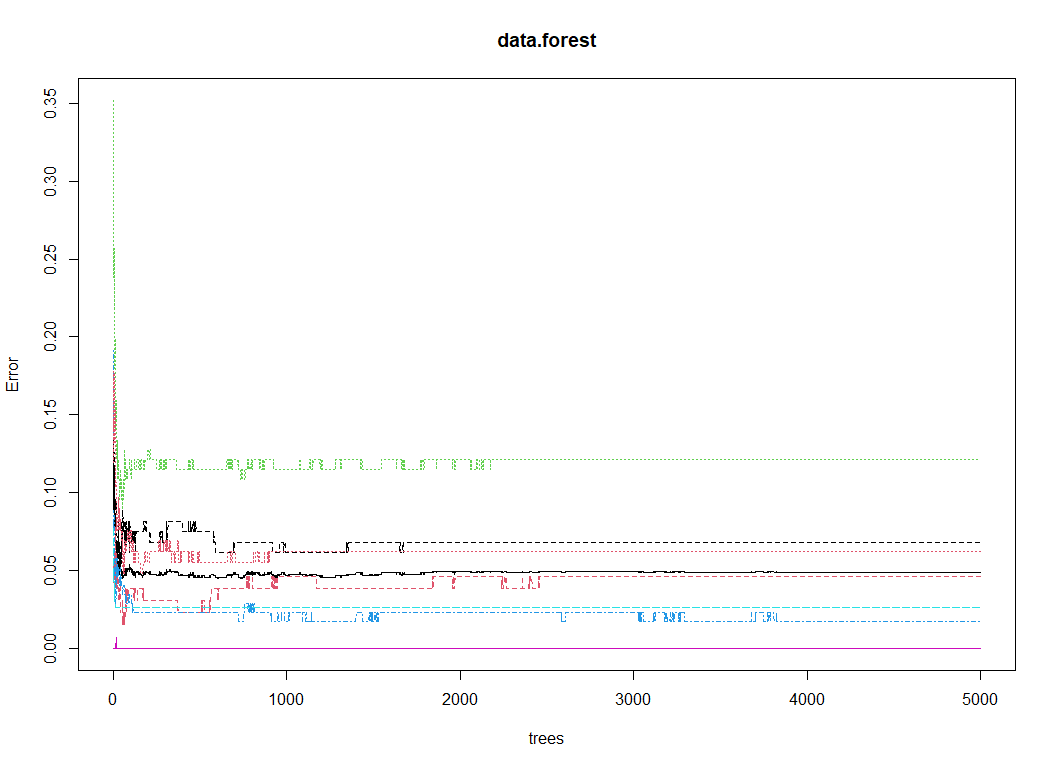
plot(data.forest)



data.forest <- randomForest(  
 weight\_category ~ .,  
 data = data,  
 subset = train,  
 ntree = 5000,  
 mtry = ncol(data) - 1,  
 importance = T,  
 localImp = T  
)  
data.forest

##   
## Call:  
## randomForest(formula = weight\_category ~ ., data = data, ntree = 5000, mtry = ncol(data) - 1, importance = T, localImp = T, subset = train)   
## Type of random forest: classification  
## Number of trees: 5000  
## No. of variables tried at each split: 16  
##   
## OOB estimate of error rate: 4.83%  
## Confusion matrix:  
## Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Insufficient\_Weight 125 6 0  
## Normal\_Weight 8 138 0  
## Obesity\_Type\_I 0 0 171  
## Obesity\_Type\_II 0 0 4  
## Obesity\_Type\_III 0 0 0  
## Overweight\_Level\_I 0 5 0  
## Overweight\_Level\_II 0 0 8  
## Obesity\_Type\_II Obesity\_Type\_III Overweight\_Level\_I  
## Insufficient\_Weight 0 0 0  
## Normal\_Weight 0 0 11  
## Obesity\_Type\_I 1 0 0  
## Obesity\_Type\_II 150 0 0  
## Obesity\_Type\_III 0 147 0  
## Overweight\_Level\_I 0 0 137  
## Overweight\_Level\_II 0 0 1  
## Overweight\_Level\_II class.error  
## Insufficient\_Weight 0 0.04580153  
## Normal\_Weight 0 0.12101911  
## Obesity\_Type\_I 2 0.01724138  
## Obesity\_Type\_II 0 0.02597403  
## Obesity\_Type\_III 0 0.00000000  
## Overweight\_Level\_I 5 0.06802721  
## Overweight\_Level\_II 136 0.06206897

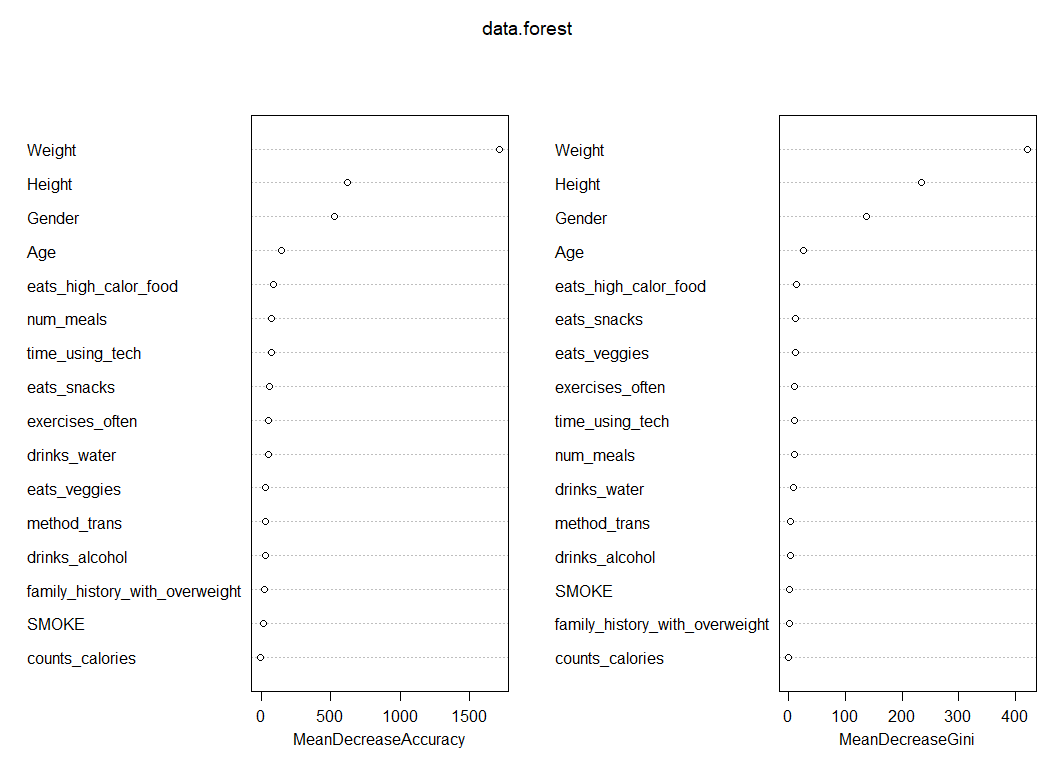
plot(data.forest)



importance(data.forest)

## Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Gender 33.567941 6.620219 1.966151e+02  
## Age 6.955835 21.969044 4.558723e+01  
## Height 192.270758 181.846909 3.600960e+02  
## Weight 789.845721 510.058705 7.635606e+02  
## family\_history\_with\_overweight 7.407269 5.099017 1.626893e+01  
## eats\_high\_calor\_food 5.663099 9.640023 2.234950e+01  
## eats\_veggies 25.962026 23.673422 2.407553e+01  
## num\_meals 43.273233 -1.686130 4.644282e+01  
## eats\_snacks 12.713725 27.770485 2.584563e+01  
## SMOKE 5.510698 7.371416 3.375441e-03  
## drinks\_water 17.480593 13.705093 3.235806e+01  
## counts\_calories -1.391688 -12.046501 2.229943e+00  
## exercises\_often 33.518625 2.515008 2.342145e+01  
## time\_using\_tech 56.256883 13.225996 3.021962e+01  
## drinks\_alcohol 4.976494 -5.180093 1.768500e+01  
## method\_trans 21.034882 2.990803 1.866125e+01  
## Obesity\_Type\_II Obesity\_Type\_III  
## Gender 399.435949 524.223206  
## Age 130.074372 6.036538  
## Height 91.795356 9.483723  
## Weight 836.583657 741.701388  
## family\_history\_with\_overweight 1.147465 0.000000  
## eats\_high\_calor\_food -1.054018 0.000000  
## eats\_veggies 12.526743 8.491329  
## num\_meals 18.060959 3.542797  
## eats\_snacks -4.620543 1.994911  
## SMOKE 0.000000 0.000000  
## drinks\_water 33.162233 4.906180  
## counts\_calories 0.000000 0.000000  
## exercises\_often 8.564850 4.189220  
## time\_using\_tech 9.752414 2.054273  
## drinks\_alcohol 18.770724 5.119241  
## method\_trans 1.732390 0.000000  
## Overweight\_Level\_I Overweight\_Level\_II  
## Gender 8.283746 35.6772526  
## Age 54.381308 49.4484004  
## Height 351.224199 410.1915457  
## Weight 633.261286 621.7230798  
## family\_history\_with\_overweight 9.362110 21.8945713  
## eats\_high\_calor\_food 8.027086 96.6361152  
## eats\_veggies 29.514952 21.8417121  
## num\_meals 31.585884 39.7168317  
## eats\_snacks 47.361689 29.0465217  
## SMOKE 19.167148 0.6177717  
## drinks\_water 15.906580 21.0812491  
## counts\_calories 10.407824 4.2376662  
## exercises\_often 34.580933 22.4536661  
## time\_using\_tech 16.347481 41.1421046  
## drinks\_alcohol 22.030244 23.1800904  
## method\_trans 20.180640 20.1115086  
## MeanDecreaseAccuracy MeanDecreaseGini  
## Gender 524.710784 137.2085709  
## Age 145.610809 25.7454345  
## Height 622.229227 233.9262225  
## Weight 1712.727676 420.7951060  
## family\_history\_with\_overweight 27.743069 1.4980195  
## eats\_high\_calor\_food 91.654534 13.1570509  
## eats\_veggies 34.824225 11.0976222  
## num\_meals 74.217117 9.8560343  
## eats\_snacks 60.636447 11.1964591  
## SMOKE 18.607139 1.5182872  
## drinks\_water 53.736385 8.6286447  
## counts\_calories -2.851231 0.4118914  
## exercises\_often 53.937537 10.1815509  
## time\_using\_tech 71.532712 9.9115028  
## drinks\_alcohol 32.608880 3.5248518  
## method\_trans 34.332685 3.7909938

varImpPlot(data.forest)



data.forest.pred <- predict(data.forest, test, typr = "class")  
table(data.forest.pred, test$weight\_category)

##   
## data.forest.pred Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Insufficient\_Weight 135 5 0  
## Normal\_Weight 6 112 0  
## Obesity\_Type\_I 0 0 169  
## Obesity\_Type\_II 0 0 5  
## Obesity\_Type\_III 0 0 0  
## Overweight\_Level\_I 0 13 0  
## Overweight\_Level\_II 0 0 3  
##   
## data.forest.pred Obesity\_Type\_II Obesity\_Type\_III Overweight\_Level\_I  
## Insufficient\_Weight 0 0 0  
## Normal\_Weight 0 0 2  
## Obesity\_Type\_I 5 1 0  
## Obesity\_Type\_II 137 0 0  
## Obesity\_Type\_III 1 176 0  
## Overweight\_Level\_I 0 0 140  
## Overweight\_Level\_II 0 0 1  
##   
## data.forest.pred Overweight\_Level\_II  
## Insufficient\_Weight 0  
## Normal\_Weight 0  
## Obesity\_Type\_I 5  
## Obesity\_Type\_II 0  
## Obesity\_Type\_III 0  
## Overweight\_Level\_I 14  
## Overweight\_Level\_II 126

mean(data.forest.pred == test$weight\_category)

## [1] 0.9422348

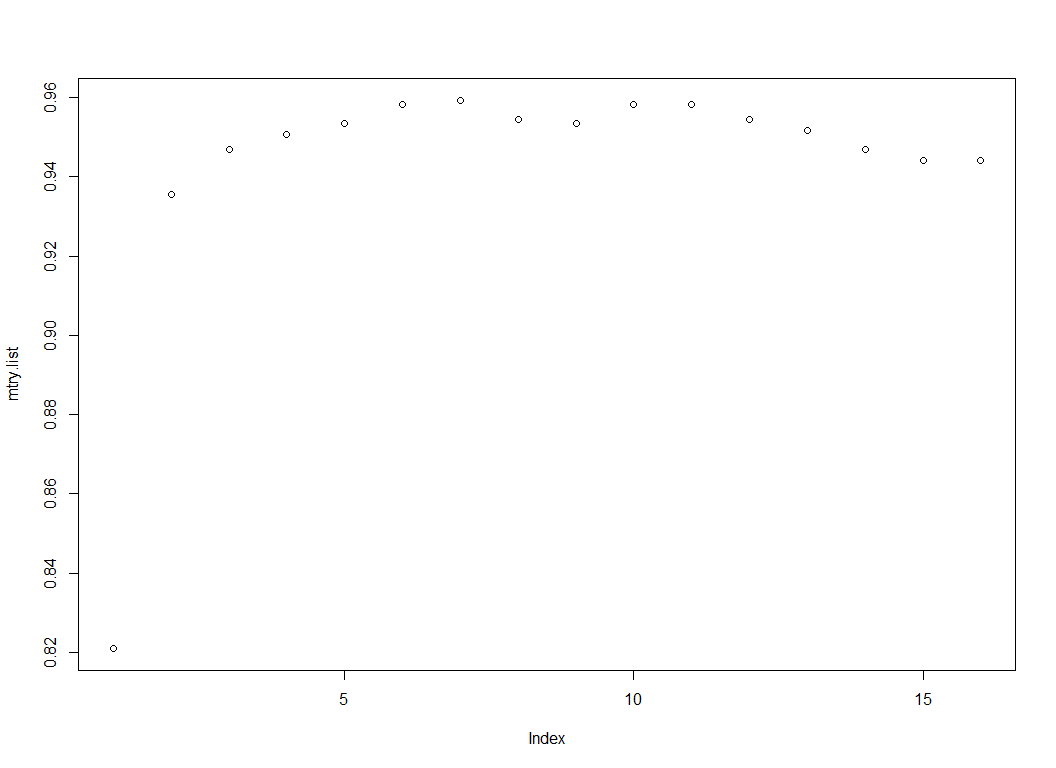
mtry.list <- c()  
data.forest.final <- NULL  
max <- 0  
for (i in 1:ncol(data) - 1) {  
 data.forest.dummy <- randomForest(  
 weight\_category ~ .,  
 data = data,  
 subset = train,  
 ntree = 5000,  
 mtry = i,  
 importance = T,  
 localImp = T  
 )  
 val <- predict(data.forest.dummy, test, typr = "class")  
 score <- mean(val == test$weight\_category)  
 if (score >= max) {  
 data.forest.final <- data.forest.dummy  
 max <- score  
 }  
 mtry.list[i] <- score  
}

## Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid  
## range

mtry.list

## [1] 0.8210227 0.9356061 0.9469697 0.9507576 0.9535985 0.9583333 0.9592803  
## [8] 0.9545455 0.9535985 0.9583333 0.9583333 0.9545455 0.9517045 0.9469697  
## [15] 0.9441288 0.9441288

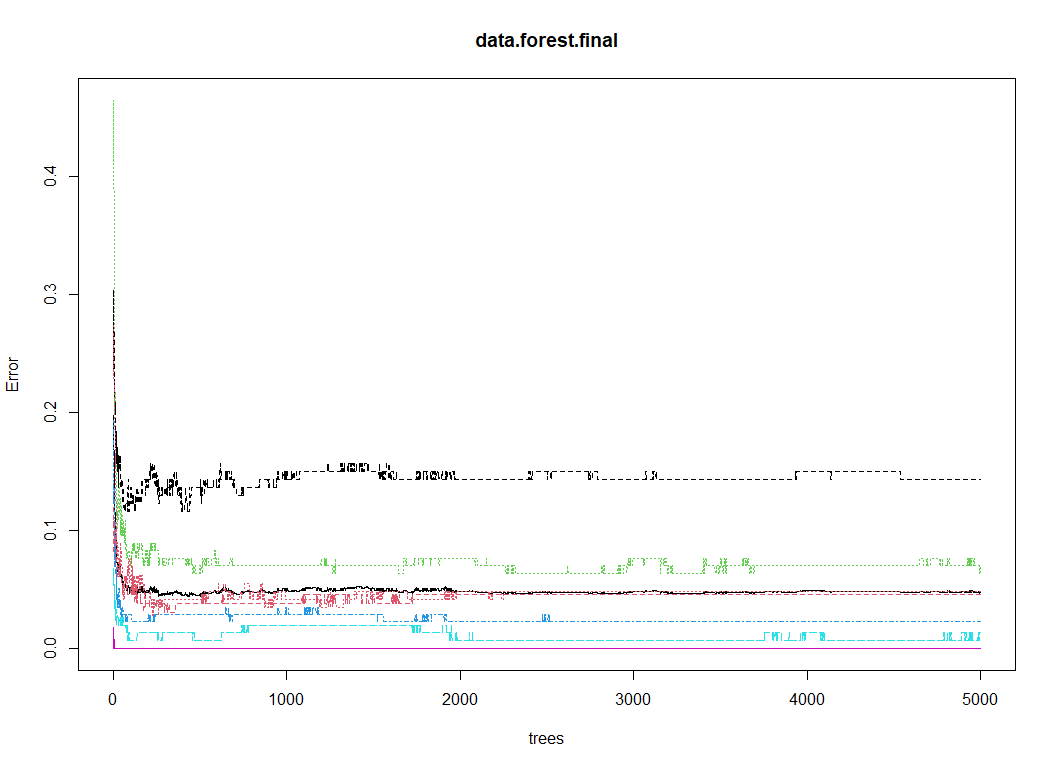
plot(mtry.list)



data.forest.final

##   
## Call:  
## randomForest(formula = weight\_category ~ ., data = data, ntree = 5000, mtry = i, importance = T, localImp = T, subset = train)   
## Type of random forest: classification  
## Number of trees: 5000  
## No. of variables tried at each split: 7  
##   
## OOB estimate of error rate: 4.74%  
## Confusion matrix:  
## Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Insufficient\_Weight 125 6 0  
## Normal\_Weight 1 147 0  
## Obesity\_Type\_I 0 0 170  
## Obesity\_Type\_II 0 1 1  
## Obesity\_Type\_III 0 0 0  
## Overweight\_Level\_I 0 15 0  
## Overweight\_Level\_II 0 1 3  
## Obesity\_Type\_II Obesity\_Type\_III Overweight\_Level\_I  
## Insufficient\_Weight 0 0 0  
## Normal\_Weight 0 0 8  
## Obesity\_Type\_I 1 0 0  
## Obesity\_Type\_II 152 0 0  
## Obesity\_Type\_III 0 147 0  
## Overweight\_Level\_I 0 0 126  
## Overweight\_Level\_II 0 0 3  
## Overweight\_Level\_II class.error  
## Insufficient\_Weight 0 0.04580153  
## Normal\_Weight 1 0.06369427  
## Obesity\_Type\_I 3 0.02298851  
## Obesity\_Type\_II 0 0.01298701  
## Obesity\_Type\_III 0 0.00000000  
## Overweight\_Level\_I 6 0.14285714  
## Overweight\_Level\_II 138 0.04827586

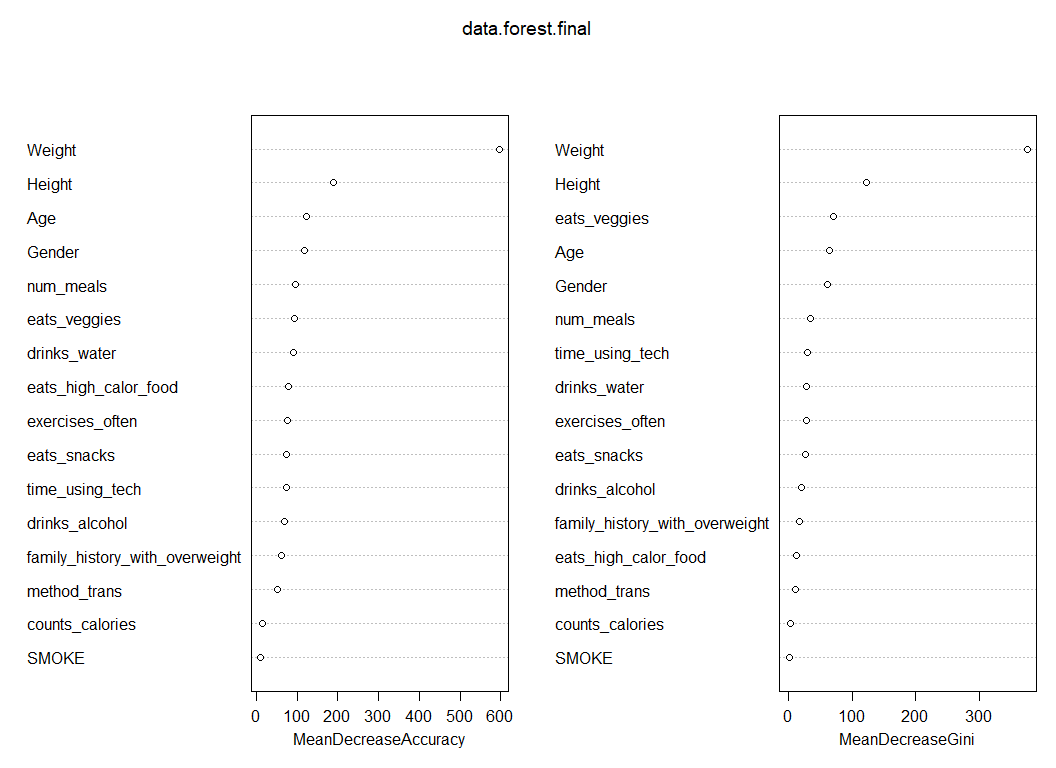
plot(data.forest.final)



importance(data.forest.final)

## Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Gender 34.092025 24.737383 75.2266425  
## Age 67.710481 28.549543 74.6069707  
## Height 83.460439 85.243075 162.2210452  
## Weight 394.303803 244.573119 365.1510592  
## family\_history\_with\_overweight 50.866284 8.281423 53.0778158  
## eats\_high\_calor\_food 14.599829 1.793716 31.5306057  
## eats\_veggies 62.171792 40.632430 68.1091819  
## num\_meals 62.804344 21.997581 61.3743698  
## eats\_snacks 56.958418 33.070499 58.4640236  
## SMOKE 6.099565 1.021447 0.6037533  
## drinks\_water 27.689033 29.853255 52.2769725  
## counts\_calories 2.105679 -14.756797 3.9420991  
## exercises\_often 36.680564 21.436526 41.7040734  
## time\_using\_tech 54.300206 25.704700 46.1003317  
## drinks\_alcohol 32.258998 -6.919627 45.2103163  
## method\_trans 40.626461 -6.427991 25.6645151  
## Obesity\_Type\_II Obesity\_Type\_III  
## Gender 70.424389 147.797763  
## Age 95.681490 30.582692  
## Height 40.474880 28.036583  
## Weight 618.538973 375.973394  
## family\_history\_with\_overweight 27.889227 25.333325  
## eats\_high\_calor\_food 10.664688 6.121156  
## eats\_veggies 69.220997 82.586299  
## num\_meals 44.396526 22.059257  
## eats\_snacks 26.420181 29.188851  
## SMOKE -1.000100 1.000100  
## drinks\_water 61.672893 10.838235  
## counts\_calories 2.558776 3.119961  
## exercises\_often 36.107660 17.062006  
## time\_using\_tech 32.183077 36.778541  
## drinks\_alcohol 41.283255 22.323652  
## method\_trans 14.371417 15.417240  
## Overweight\_Level\_I Overweight\_Level\_II  
## Gender 44.78631 52.613198  
## Age 87.33846 86.095053  
## Height 132.68427 158.402363  
## Weight 291.84756 295.533931  
## family\_history\_with\_overweight 46.04434 52.689469  
## eats\_high\_calor\_food 35.65926 80.312414  
## eats\_veggies 55.97043 56.230980  
## num\_meals 58.59219 61.267745  
## eats\_snacks 53.40308 55.861565  
## SMOKE 11.87252 3.289337  
## drinks\_water 42.23773 34.615059  
## counts\_calories 25.69223 11.296825  
## exercises\_often 48.35104 36.497466  
## time\_using\_tech 38.57782 40.534062  
## drinks\_alcohol 53.65546 58.473015  
## method\_trans 37.65273 30.386617  
## MeanDecreaseAccuracy MeanDecreaseGini  
## Gender 118.83381 60.307372  
## Age 123.97535 64.464198  
## Height 188.22917 122.089412  
## Weight 595.93457 375.588368  
## family\_history\_with\_overweight 61.46337 16.801670  
## eats\_high\_calor\_food 78.20487 12.971551  
## eats\_veggies 94.59902 70.468218  
## num\_meals 96.38532 34.225086  
## eats\_snacks 74.79345 26.171021  
## SMOKE 10.72443 1.324411  
## drinks\_water 91.62780 27.787919  
## counts\_calories 15.06997 2.204496  
## exercises\_often 77.57069 27.739343  
## time\_using\_tech 72.70208 29.294949  
## drinks\_alcohol 68.64138 19.737188  
## method\_trans 50.67879 11.130932

varImpPlot(data.forest.final)



data.forest.final.pred <-  
 predict(data.forest.final, test, typr = "class")  
table(data.forest.final.pred, test$weight\_category)

##   
## data.forest.final.pred Insufficient\_Weight Normal\_Weight Obesity\_Type\_I  
## Insufficient\_Weight 134 2 0  
## Normal\_Weight 7 124 0  
## Obesity\_Type\_I 0 0 174  
## Obesity\_Type\_II 0 0 2  
## Obesity\_Type\_III 0 0 0  
## Overweight\_Level\_I 0 4 1  
## Overweight\_Level\_II 0 0 0  
##   
## data.forest.final.pred Obesity\_Type\_II Obesity\_Type\_III Overweight\_Level\_I  
## Insufficient\_Weight 0 0 0  
## Normal\_Weight 0 0 8  
## Obesity\_Type\_I 1 1 0  
## Obesity\_Type\_II 141 0 0  
## Obesity\_Type\_III 1 176 0  
## Overweight\_Level\_I 0 0 131  
## Overweight\_Level\_II 0 0 4  
##   
## data.forest.final.pred Overweight\_Level\_II  
## Insufficient\_Weight 0  
## Normal\_Weight 5  
## Obesity\_Type\_I 2  
## Obesity\_Type\_II 0  
## Obesity\_Type\_III 0  
## Overweight\_Level\_I 5  
## Overweight\_Level\_II 133

mean(data.forest.final.pred == test$weight\_category)

## [1] 0.9592803

# explain\_forest(data.forest.final, interactions = TRUE, data = data)