practical homework - 1

```r
#Imported all the necessary libraries
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
## Warning: package 'tibble' was built under R version 4.4.2
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
## Warning: package 'purrr' was built under R version 4.4.2
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
## Warning: package 'forcats' was built under R version 4.4.2
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.4.2
```

```r
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.4.3
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.4.2

## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(gbm)
```

```
## Warning: package 'gbm' was built under R version 4.4.3

## Loaded gbm 2.2.2
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.c
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.2

## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
# I've set the working directory and loaded 'youth_data.Rdata' into data
setwd("C:/Users/alekh/Downloads/")
data <- load("youth_data.Rdata")
youth_experience_cols
```

```
##  [1] "SCHFELT"  "TCHGJOB"  "AVGGRADE" "STNDSCIG" "STNDSMJ"  "STNDALC"
##  [7] "STNDDNK"  "PARCHKHW" "PARHLPHW" "PRCHORE2" "PRLMTTV2" "PARLMTSN"
## [13] "PRGDJOB2" "PRPROUD2" "ARGUPAR"  "YOFIGHT2" "YOGRPFT2" "YOHGUN2"
## [19] "YOSELL2"  "YOSTOLE2" "YOATTAK2" "PRPKCIG2" "PRMJEVR2" "PRMJMO"
## [25] "PRALDLY2" "YFLPKCG2" "YFLTMRJ2" "YFLMJMO"  "YFLADLY2" "FRDPCIG2"
## [31] "FRDMEVR2" "FRDMJMON" "FRDADLY2" "TALKPROB" "PRTALK3"  "PRBSOLV2"
## [37] "PREVIOL2" "PRVDRGO2" "GRPCNSL2" "PREGPGM2" "YTHACT2"  "DRPRVME3"
## [43] "ANYEDUC3" "RLGATTD"  "RLGIMPT"  "RLGDCSN"  "RLGFRND"
```

```r
substance_cols
```

```
##  [1] "IRALCFY"     "IRMJFY"      "IRCIGFM"     "IRSMKLSS30N" "IRALCFM"    
##  [6] "IRMJFM"      "IRCIGAGE"    "IRSMKLSSTRY" "IRALCAGE"    "IRMJAGE"    
## [11] "MRJFLAG"     "ALCFLAG"     "TOBFLAG"     "ALCYDAYS"    "MRJYDAYS"   
## [16] "ALCMDAYS"    "MRJMDAYS"    "CIGMDAYS"    "SMKLSMDAYS"
```

```r
demographic_cols
```

```
##  [1] "IRSEX"      "NEWRACE2"   "HEALTH2"    "EDUSCHLGO"  "EDUSCHGRD2"
##  [6] "EDUSKPCOM"  "IMOTHER"    "IFATHER"    "INCOME"     "GOVTPROG"  
## [11] "POVERTY3"   "PDEN10"     "COUTYP4"
```

```r
# Named the dataframe 'df' as drug_use
drug_use <- na.omit(df)

# PART1: BINARY CLASSIFICATION: Predicting whether a youth has ever consumed alcohol or not
# The dataframe 'df_all' consists of the predictors and target variable
df_all <- drug_use[, c(demographic_cols, youth_experience_cols, "ALCFLAG")]
df_all <- na.omit(df_all)

df_all$alcohol_use <- factor(df_all$ALCFLAG, levels = c(0, 1), labels = c("No", "Yes"))
df_all$ALCFLAG <- NULL


# To readability, I'm renaming the predictors
colnames(df_all)[colnames(df_all) == "STNDALC"]     <- "Friend_Drinks_Daily"
colnames(df_all)[colnames(df_all) == "YFLMJMO"]     <- "Friend_Consumes_Marijuana_Monthly"
colnames(df_all)[colnames(df_all) == "YFLTMRJ2"]    <- "Friend_Offers_Marijuana"
colnames(df_all)[colnames(df_all) == "FRDMEVR2"]    <- "Friend_Ever_Smoked"
colnames(df_all)[colnames(df_all) == "STNDSMJ"]     <- "Friend_Smokes_Marijuana"
colnames(df_all)[colnames(df_all) == "EDUSCHGRD2"]  <- "Grade_Level"
colnames(df_all)[colnames(df_all) == "NEWRACE2"]    <- "Race"

# For plot readability, recoding categorical variables.
df_all$Friend_Drinks_Daily <- factor(df_all$Friend_Drinks_Daily, levels = c(1, 2), labels = c("Yes", "N

df_all$Friend_Consumes_Marijuana_Monthly <- factor(df_all$Friend_Consumes_Marijuana_Monthly, levels = c

df_all$Friend_Offers_Marijuana <- factor(df_all$Friend_Offers_Marijuana, levels = c(1, 2), labels = c("

# Plotting the decision tree
tree_one <- tree(alcohol_use ~ ., data = df_all)
tree_one
```

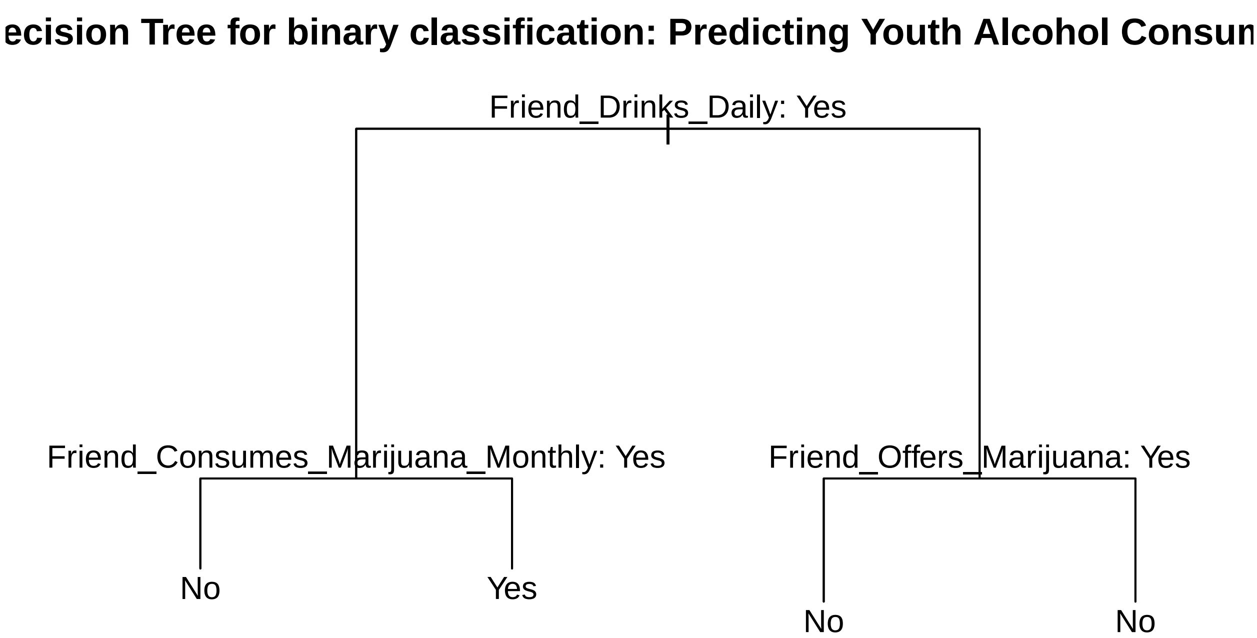
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
## 
## 1) root 8249 9250 No ( 0.7515 0.2485 )  
##   2) Friend_Drinks_Daily: Yes 2366 3279 No ( 0.5118 0.4882 )  
##     4) Friend_Consumes_Marijuana_Monthly: Yes 1435 1875 No ( 0.6404 0.3596 ) *
##     5) Friend_Consumes_Marijuana_Monthly: No 931 1158 Yes ( 0.3136 0.6864 ) *
```

```
##   3) Friend_Drinks_Daily: No 5883 5017 No ( 0.8479 0.1521 )
##     6) Friend_Offers_Marijuana: Yes 4885 3376 No ( 0.8905 0.1095 ) *
##     7) Friend_Offers_Marijuana: No 998 1305 No ( 0.6393 0.3607 ) *
```

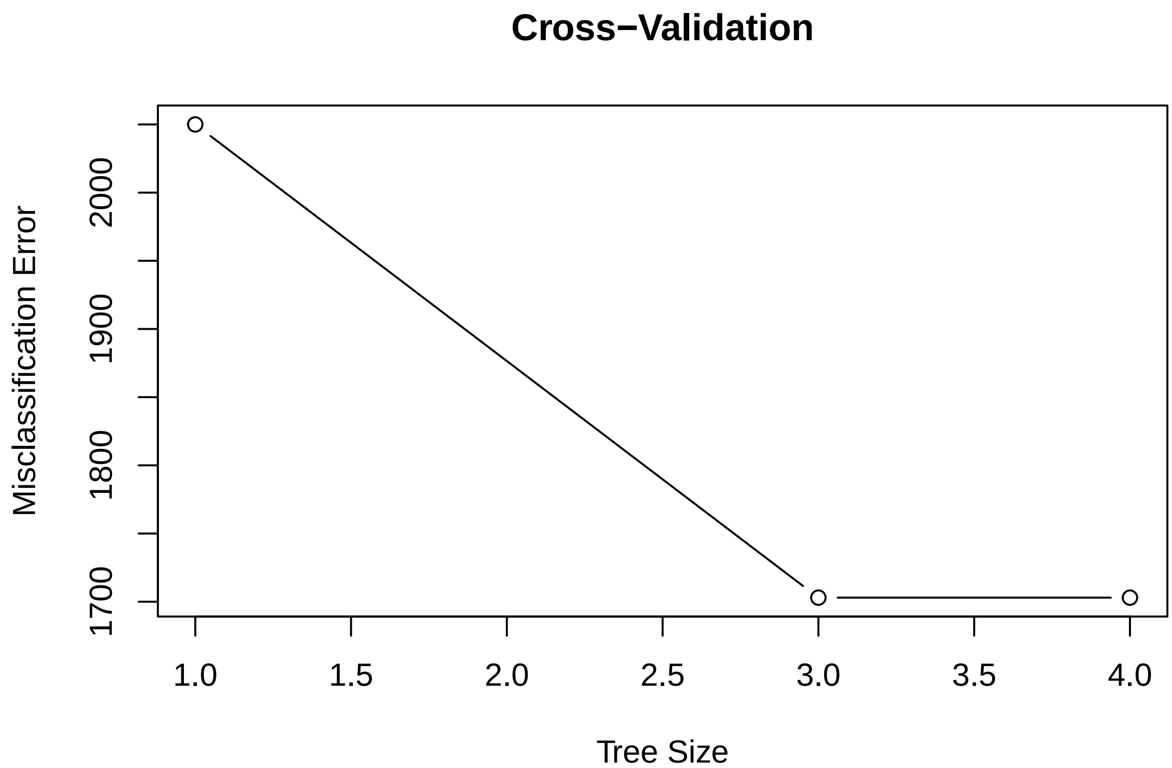
```r
summary(tree_one)
```

```
##
## Classification tree:
## tree(formula = alcohol_use ~ ., data = df_all)
## Variables actually used in tree construction:
## [1] "Friend_Drinks_Daily"               "Friend_Consumes_Marijuana_Monthly"
## [3] "Friend_Offers_Marijuana"
## Number of terminal nodes:  4
## Residual mean deviance:  0.9355 = 7713 / 8245
## Misclassification error rate: 0.2064 = 1703 / 8249
```

```r
plot(tree_one)
text(tree_one, pretty = 0)
title("Decision Tree for binary classification: Predicting Youth Alcohol Consumption")
```



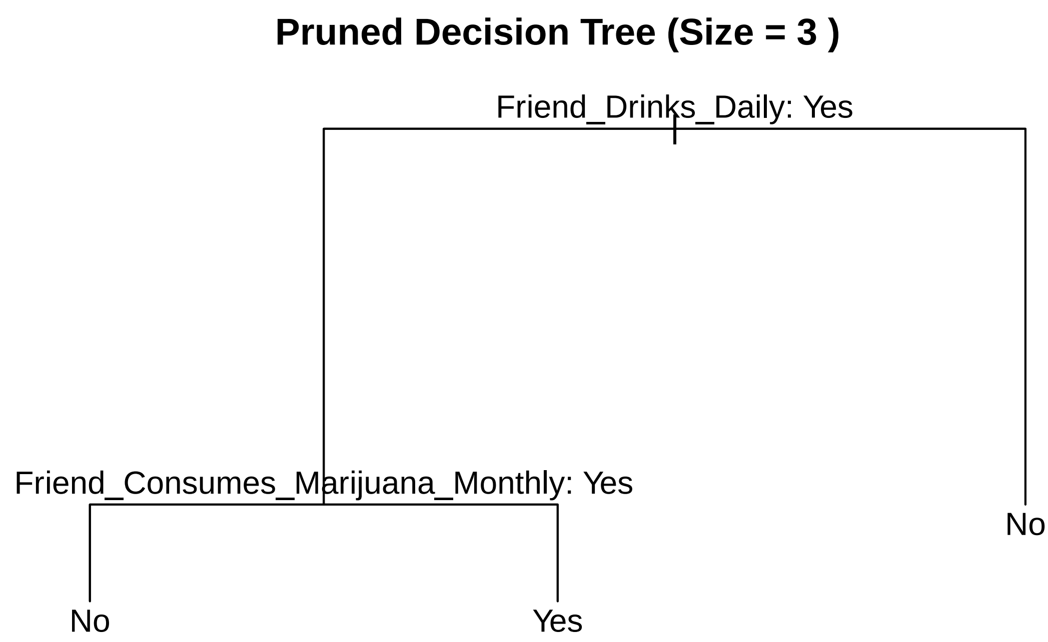

```r
# Pruning the above tree
# Finding optimal size, using cross validation
set.seed(1)
cv_one <- cv.tree(tree_one, FUN = prune.misclass)
plot(cv_one$size, cv_one$dev, type = "b", main = "Cross-Validation", xlab = "Tree Size", ylab = "Miscla
```




```
opt_size <- cv_one$size[which.min(cv_one$dev)]
opt_size
```

```
## [1] 4
```

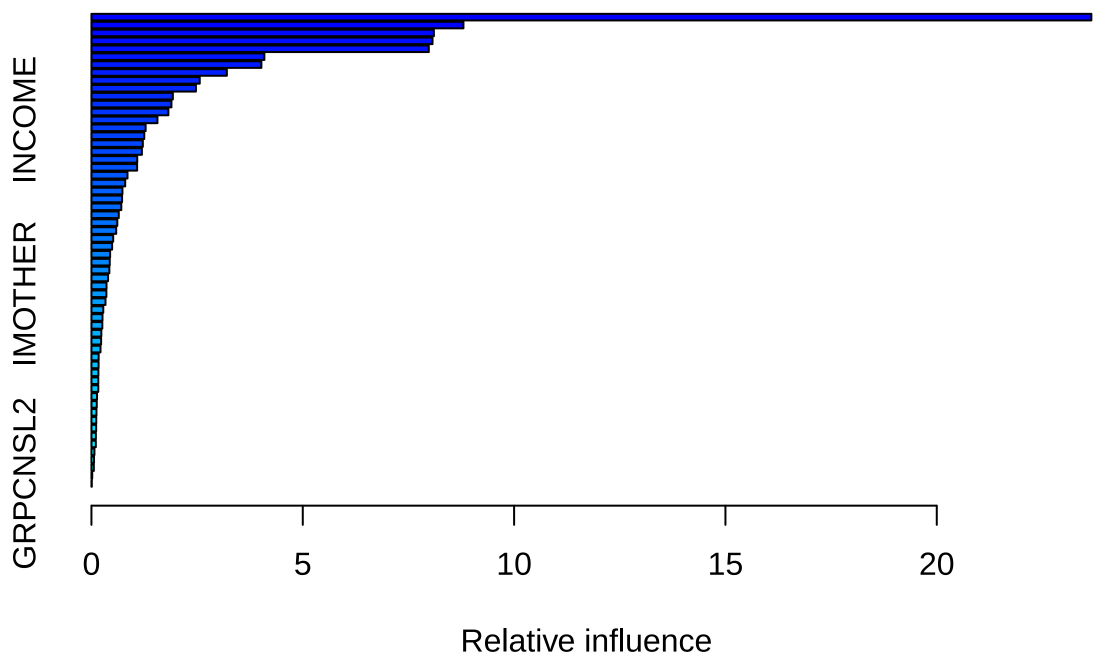
```
# We got best size as 4
# But, 3 is the best optimal size from the graph
# So, we use 3 as the opt_size
prune_one <- prune.misclass(tree_one, best = 3)
plot(prune_one)
text(prune_one, pretty = 0)
title(paste("Pruned Decision Tree (Size =", 3, ")"))
```




```r
# DECISION TREE ENSEMBLE METHODS

# Getting the numeric version of alcohol_use(categorical)
df_all$alcohol_use_num <- ifelse(df_all$alcohol_use == "Yes", 1, 0)

# Splitting the data into train and test data.
set.seed(42)
train_data <- sample(1:nrow(df_all), 0.6 * nrow(df_all))
train_set <- df_all[train_data, ]
test_set  <- df_all[-train_data, ]

# BAGGING
bag_one <- randomForest(alcohol_use ~ ., data = train_set[, -which(names(train_set) == "alcohol_use_num
prediction_bag <- predict(bag_one, test_set)
cat("Bagging Accuracy:", mean(prediction_bag == test_set$alcohol_use), "\n")
```

```
## Bagging Accuracy: 0.7915152
```

```r
# BOOSTING
set.seed(1)
boost_one <- gbm(alcohol_use_num ~ ., data = train_set[, -which(names(train_set) == "alcohol_use")], di
summary(boost_one)
```




```
##                                                                   var
## Friend_Drinks_Daily                               Friend_Drinks_Daily
## Friend_Offers_Marijuana                       Friend_Offers_Marijuana
## FRDMJMON                                                     FRDMJMON
## Friend_Consumes_Marijuana_Monthly Friend_Consumes_Marijuana_Monthly
## Grade_Level                                               Grade_Level
## YOSTOLE2                                                     YOSTOLE2
## Race                                                             Race
## Friend_Smokes_Marijuana                       Friend_Smokes_Marijuana
## PRMJEVR2                                                     PRMJEVR2
## PRALDLY2                                                     PRALDLY2
## POVERTY3                                                     POVERTY3
## FRDPCIG2                                                     FRDPCIG2
## YOFIGHT2                                                     YOFIGHT2
## INCOME                                                         INCOME
## Friend_Ever_Smoked                                 Friend_Ever_Smoked
## YOHGUN2                                                       YOHGUN2
## YTHACT2                                                       YTHACT2
## ARGUPAR                                                       ARGUPAR
## PARHLPHW                                                     PARHLPHW
## PRLMTTV2                                                     PRLMTTV2
## RLGDCSN                                                       RLGDCSN
## HEALTH2                                                       HEALTH2
## IRSEX                                                           IRSEX
## EDUSKPCOM                                                   EDUSKPCOM
## PRGDJOB2                                                     PRGDJOB2
```

```
## YFLPKCG2                                                YFLPKCG2
## YOATTAK2                                                YOATTAK2
## YOGRPFT2                                                YOGRPFT2
## YOSELL2                                                  YOSELL2
## COUTYP4                                                  COUTYP4
## PARCHKHW                                                PARCHKHW
## PRPROUD2                                                PRPROUD2
## STNDSCIG                                                STNDSCIG
## YFLADLY2                                                YFLADLY2
## TCHGJOB                                                  TCHGJOB
## IMOTHER                                                  IMOTHER
## PARLMTSN                                                PARLMTSN
## EDUSCHLGO                                              EDUSCHLGO
## PRBSOLV2                                                PRBSOLV2
## PRTALK3                                                  PRTALK3
## PRMJMO                                                    PRMJMO
## PRPKCIG2                                                PRPKCIG2
## FRDADLY2                                                FRDADLY2
## STNDDNK                                                  STNDDNK
## ANYEDUC3                                                ANYEDUC3
## RLGIMPT                                                  RLGIMPT
## GOVTPROG                                                GOVTPROG
## RLGATTD                                                  RLGATTD
## PDEN10                                                    PDEN10
## IFATHER                                                  IFATHER
## DRPRVME3                                                DRPRVME3
## PREVIOL2                                                PREVIOL2
## PRCHORE2                                                PRCHORE2
## TALKPROB                                                TALKPROB
## RLGFRND                                                  RLGFRND
## SCHFELT                                                  SCHFELT
## PREGPGM2                                                PREGPGM2
## AVGGRADE                                                AVGGRADE
## PRVDRGO2                                                PRVDRGO2
## GRPCNSL2                                                GRPCNSL2
##                                        rel.inf
## Friend_Drinks_Daily                23.657686794
## Friend_Offers_Marijuana             8.801175627
## FRDMJMON                            8.101931778
## Friend_Consumes_Marijuana_Monthly   8.069717743
## Grade_Level                         7.981885230
## YOSTOLE2                            4.091588461
## Race                                4.021069005
## Friend_Smokes_Marijuana             3.203816039
## PRMJEVR2                            2.562217942
## PRALDLY2                            2.472109375
## POVERTY3                            1.926140517
## FRDPCIG2                            1.892961009
## YOFIGHT2                            1.820244286
## INCOME                              1.561601333
## Friend_Ever_Smoked                  1.282025751
## YOHGUN2                             1.249758413
## YTHACT2                             1.215895993
## ARGUPAR                             1.193683726
```

```
## PARHLPHW                        1.083686698
## PRLMTTV2                        1.080762218
## RLGDCSN                         0.853272848
## HEALTH2                         0.798697620
## IRSEX                           0.733340070
## EDUSKPCOM                       0.724825601
## PRGDJOB2                        0.706366412
## YFLPKCG2                        0.646551711
## YOATTAK2                        0.611941392
## YOGRPFT2                        0.585497902
## YOSELL2                         0.514596973
## COUTYP4                         0.488768559
## PARCHKHW                        0.441477934
## PRPROUD2                        0.433268764
## STNDSCIG                        0.424011249
## YFLADLY2                        0.392808064
## TCHGJOB                         0.356282457
## IMOTHER                         0.353311754
## PARLMTSN                        0.332352492
## EDUSCHLGO                       0.279930255
## PRBSOLV2                        0.261190771
## PRTALK3                         0.257618502
## PRMJMO                          0.233986255
## PRPKCIG2                        0.228922485
## FRDADLY2                        0.214487655
## STNDDNK                         0.171029926
## ANYEDUC3                        0.168874024
## RLGIMPT                         0.164505715
## GOVTPROG                        0.162452863
## RLGATTD                         0.161975122
## PDEN10                          0.132673861
## IFATHER                         0.126337299
## DRPRVME3                        0.118827566
## PREVIOL2                        0.113939032
## PRCHORE2                        0.111964945
## TALKPROB                        0.104709265
## RLGFRND                         0.103424030
## SCHFELT                         0.070642171
## PREGPGM2                        0.061219728
## AVGGRADE                        0.056538813
## PRVDRGO2                        0.018595891
## GRPCNSL2                        0.008824084
```

```r
probability_one <- predict(boost_one, test_set, n.trees = 1000, type = "response")
prediction_boost <- ifelse(probability_one > 0.5, "Yes", "No")
cat("Boosting Accuracy:", mean(prediction_boost == test_set$alcohol_use), "\n")
```

```
## Boosting Accuracy: 0.8069697
```

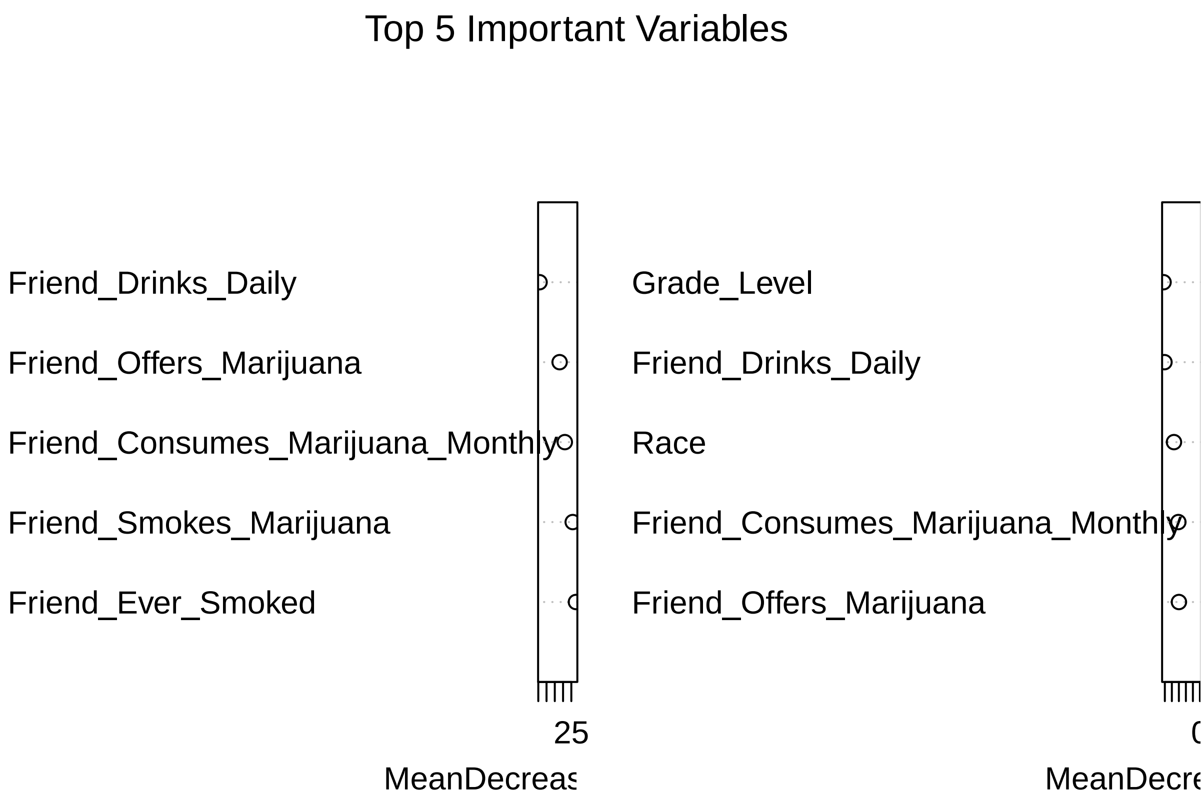
```r
# RANDOM FOREST
rf_one <- randomForest(alcohol_use ~ ., data = train_set[, -which(names(train_set) == "alcohol_use_num"
# mtry = 5 since we took sqrt(p)
prediction_rf <- predict(rf_one, test_set)
confusionMatrix(prediction_rf, test_set$alcohol_use)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  2337  523
##        Yes  139  301
##
##                Accuracy : 0.7994
##                  95% CI : (0.7853, 0.8129)
##     No Information Rate : 0.7503
##     P-Value [Acc > NIR] : 1.471e-11
##
##                   Kappa : 0.3661
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9439
##             Specificity : 0.3653
##          Pos Pred Value : 0.8171
##          Neg Pred Value : 0.6841
##              Prevalence : 0.7503
##          Detection Rate : 0.7082
##    Detection Prevalence : 0.8667
##       Balanced Accuracy : 0.6546
##
##        'Positive' Class : No
##
```

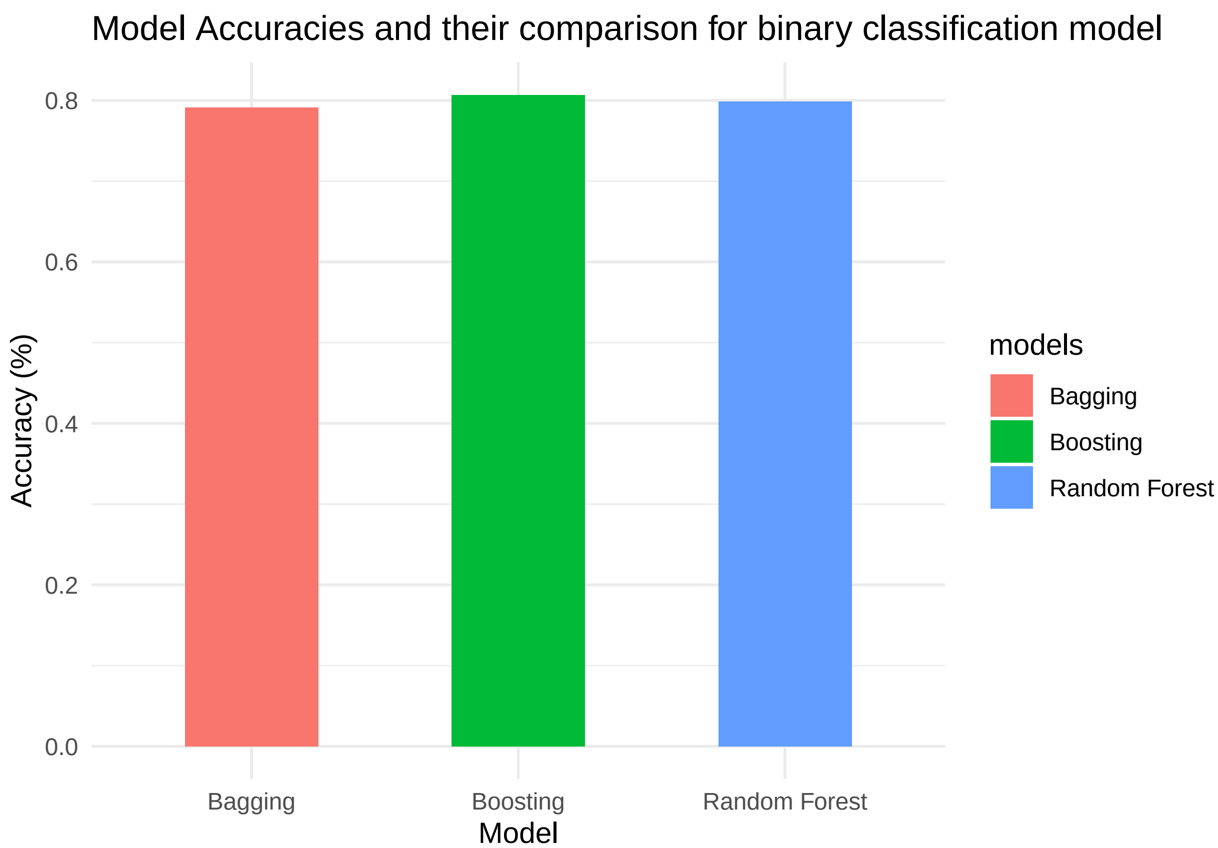
```r
cat("Random Forest Accuracy:", mean(prediction_rf == test_set$alcohol_use), "\n")
```

```
## Random Forest Accuracy: 0.7993939
```

```r
varImpPlot(rf_one, n.var = 5, sort = TRUE, main = "Top 5 Important Variables")
```




```r
# plotting the three models accuracy results
models <- c("Boosting", "Random Forest", "Bagging")
accuracies <- c(80.7, 79.9, 79.1)/100
df_accuracies <- data.frame(models, accuracies)
ggplot(df_accuracies, aes(x = models, y = accuracies, fill = models)) +
  geom_col(width = 0.5, show.legend = TRUE) +
  labs(title = "Model Accuracies and their comparison for binary classification model",
       x = "Model", y = "Accuracy (%)") +
  theme_minimal()
```



Ensemble models were compared using accuracy: Boosting: 80.7% Random Forest: 79.9% Bagging: 79.1%

And from the variable plot, Friend Drinks Daily, Friend Offers Marijuana, and Grade Level are the most important predictors identified by the model

```
# PART 2
# MULTI-CLASS CLASSIFICATION: how often they used marijuana over the last year

# Converting MRJYDAYS to numeric and also performing some data cleaning by including data, like these:
drug_use$MRJYDAYS <- as.numeric(as.character(drug_use$MRJYDAYS))

# Imputation with mean for 900-level codes
invalid_codes <- c(991, 993, 994, 997, 998)
valid_mean <- mean(drug_use$MRJYDAYS[!drug_use$MRJYDAYS %in% invalid_codes], na.rm = TRUE)
drug_use$MRJYDAYS[drug_use$MRJYDAYS %in% invalid_codes] <- valid_mean

# Now, binning the data into 6 categories
drug_use$marijuana_use_level <- cut(
  drug_use$MRJYDAYS,
  breaks = c(-1, 0, 5, 15, 30, 90, 365),
  labels = c("NEVER", "RARE", "OCCASIONAL", "REGULAR", "FREQUENT", "DAILY"),
  right = TRUE
)

# The dataframe 'df_multi' consists of the predictors and target variable for multi-classification
df_multi <- drug_use[, c(demographic_cols, youth_experience_cols, "marijuana_use_level")]
df_multi <- na.omit(df_multi)
```

```r
# Renaming variables for clarity
colnames(df_multi)[colnames(df_multi) == "FRDMJMON"] <- "Friend_Uses_Marijuana_Monthly"
colnames(df_multi)[colnames(df_multi) == "STNDSMJ"]  <- "Friend_Smokes_Marijuana"
colnames(df_multi)[colnames(df_multi) == "YFLMJMO"]    <- "Friend_Consumes_Marijuana_Monthly"
colnames(df_multi)[colnames(df_multi) == "FRDMEVR2"]   <- "Friend_Ever_Tried_Marijuana"
colnames(df_multi)[colnames(df_multi) == "YOSELL2"]    <- "Youth_Sold_Drugs"
colnames(df_multi)[colnames(df_multi) == "EDUSCHGRD2"] <- "Grade_Level"
colnames(df_multi)[colnames(df_multi) == "YFLTMRJ2"]   <- "Friend_Offered_Marijuana"
colnames(df_multi)[colnames(df_multi) == "NEWRACE2"]   <- "Race"

# For plot readability, recoding categorical variables.
df_multi$Friend_Uses_Marijuana_Monthly <- factor(df_multi$Friend_Uses_Marijuana_Monthly, levels = c(1, 

df_multi$Friend_Smokes_Marijuana <- factor(df_multi$Friend_Smokes_Marijuana, levels = c(1, 2), labels =

df_multi$Friend_Consumes_Marijuana_Monthly <- factor(df_multi$Friend_Consumes_Marijuana_Monthly, levels

df_multi$Friend_Ever_Tried_Marijuana <- factor(df_multi$Friend_Ever_Tried_Marijuana, levels = c(1, 2), 

df_multi$Youth_Sold_Drugs <- factor(df_multi$Youth_Sold_Drugs, levels = c(1, 2), labels = c("Yes", "No")

df_multi$Grade_Level <- factor(df_multi$Grade_Level, levels = c(1:8, 9:11, 98, 99),
                               labels = c(
                                 rep("School", 8),
                                 rep("College", 3),
                                 "No Answer", "Skipped"
                               )
)

df_multi$Friend_Offered_Marijuana <- factor(df_multi$Friend_Offered_Marijuana, levels = c(1, 2), labels

# Plotting the decision tree
tree_two <- tree(marijuana_use_level ~ ., data = df_multi)
tree_two
```

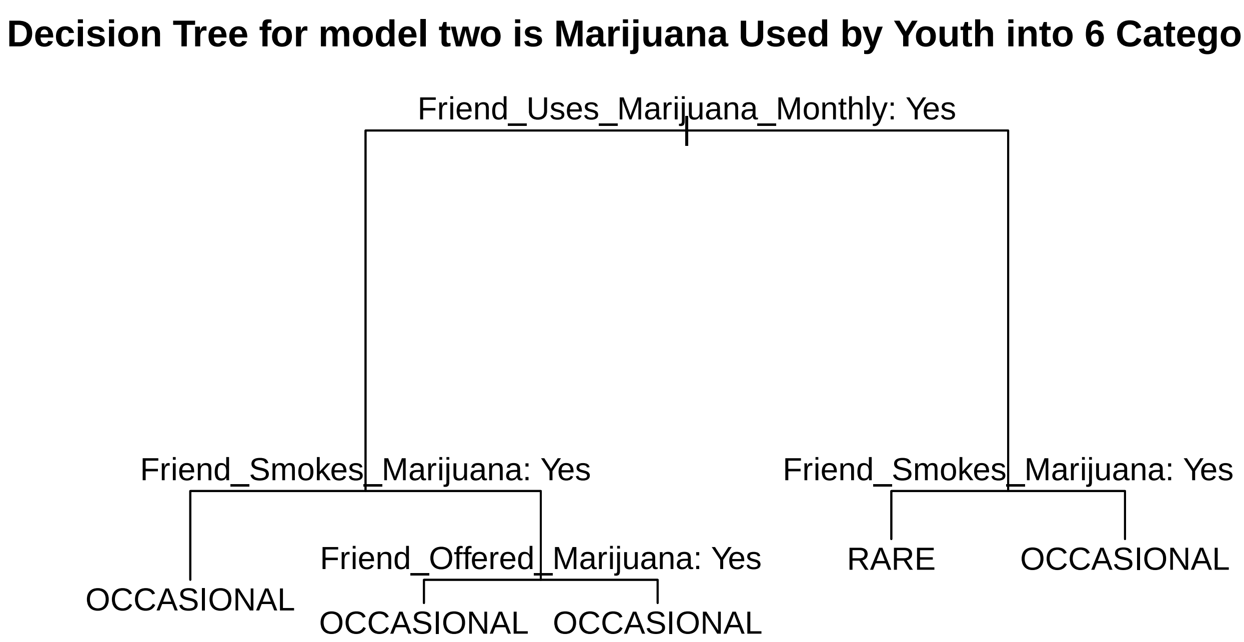
```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 8249 6583.0 OCCASIONAL ( 0.00000 0.13674 0.86326 0.00000 0.00000 0.00000 )
##    2) Friend_Uses_Marijuana_Monthly: Yes 6400 2738.0 OCCASIONAL ( 0.00000 0.05531 0.94469 0.00000 0.0
##      4) Friend_Smokes_Marijuana: Yes 1300 1192.0 OCCASIONAL ( 0.00000 0.17154 0.82846 0.00000 0.0000
##      5) Friend_Smokes_Marijuana: No 5100 1218.0 OCCASIONAL ( 0.00000 0.02569 0.97431 0.00000 0.00000
##       10) Friend_Offered_Marijuana: Yes 4681  833.9 OCCASIONAL ( 0.00000 0.01773 0.98227 0.00000 0.0
##       11) Friend_Offered_Marijuana: No 419  298.3 OCCASIONAL ( 0.00000 0.11456 0.88544 0.00000 0.000
##    3) Friend_Uses_Marijuana_Monthly: No 1849 2514.0 OCCASIONAL ( 0.00000 0.41860 0.58140 0.00000 0.0
##      6) Friend_Smokes_Marijuana: Yes 1077 1484.0 RARE ( 0.00000 0.54596 0.45404 0.00000 0.00000 0.00
##      7) Friend_Smokes_Marijuana: No 772  852.5 OCCASIONAL ( 0.00000 0.24093 0.75907 0.00000 0.00000 
```

```r
summary(tree_two)
```

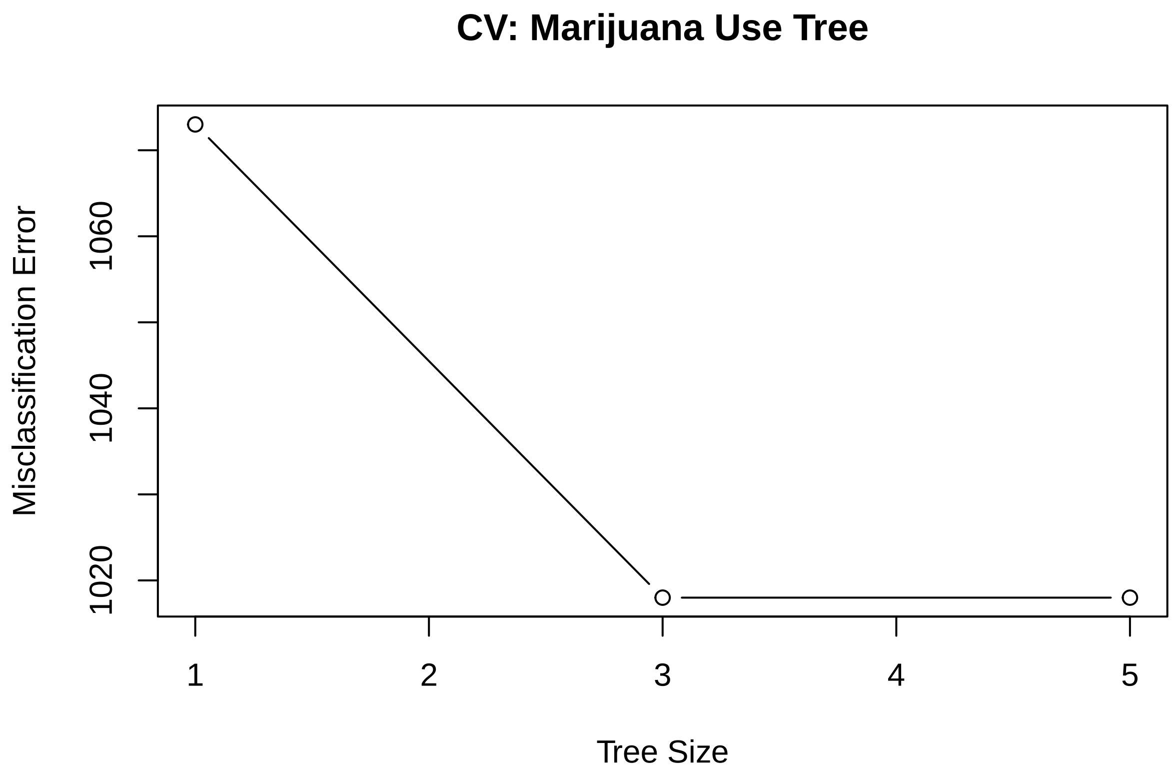
```
##
## Classification tree:
```

```
## tree(formula = marijuana_use_level ~ ., data = df_multi)
## Variables actually used in tree construction:
## [1] "Friend_Uses_Marijuana_Monthly" "Friend_Smokes_Marijuana"
## [3] "Friend_Offered_Marijuana"
## Number of terminal nodes:  5
## Residual mean deviance:  0.5653 = 4660 / 8244
## Misclassification error rate: 0.1247 = 1029 / 8249
```

```r
plot(tree_two)
text(tree_two, pretty = 0)
title("Decision Tree for model two is Marijuana Used by Youth into 6 Categories")
```



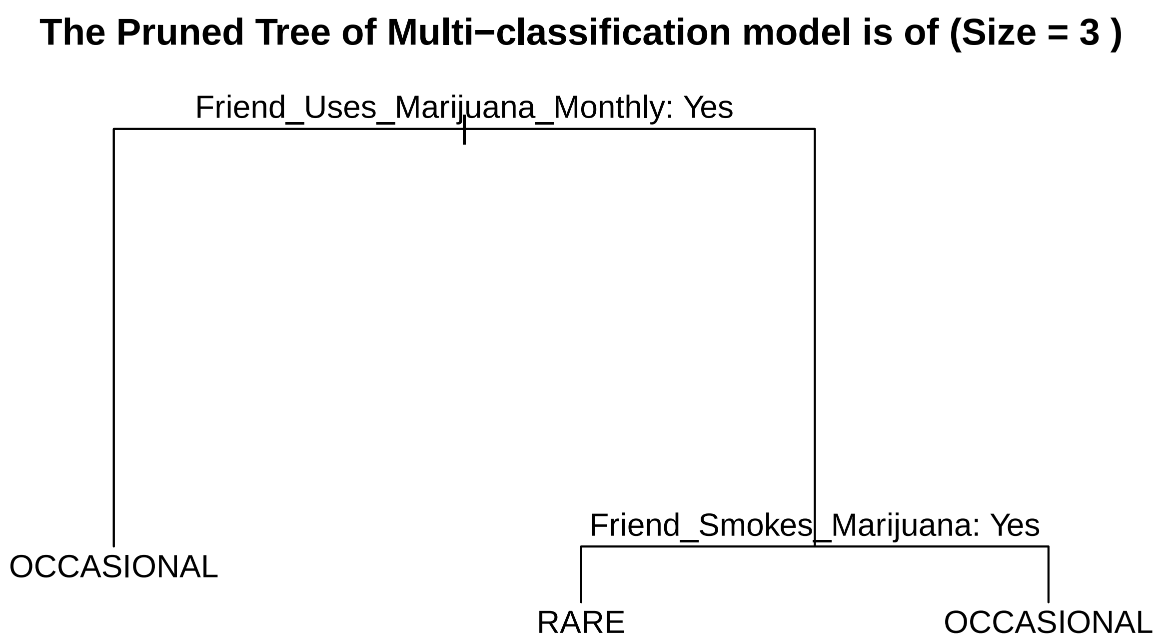

```r
# Pruning the above tree
# Finding optimal tree size, we use cross validation
set.seed(1)
cv_two <- cv.tree(tree_two, FUN = prune.misclass)
plot(cv_two$size, cv_two$dev, type = "b", main = "CV: Marijuana Use Tree", xlab = "Tree Size", ylab = "I
```




```r
opt_size <- cv_two$size[which.min(cv_two$dev)]
opt_size
```

```
## [1] 5
```

```r
# 3 is the best optimal size from the graph
# So, we use 3 as the opt_size
prune_two <- prune.misclass(tree_two, best = 3)
plot(prune_two)
text(prune_two, pretty = 0)
title(paste("The Pruned Tree of Multi-classification model is of (Size =", 3, ")"))
```




```r
# DECISION TREE ENSEMBLE METHODS
# Splitting the data into train and test data.
set.seed(123)
train_data <- createDataPartition(df_multi$marijuana_use_level, p = 0.8, list = FALSE)
```

```
## Warning in createDataPartition(df_multi$marijuana_use_level, p = 0.8, list =
## FALSE): Some classes have no records ( NEVER, REGULAR, FREQUENT, DAILY ) and
## these will be ignored
```

```r
train_set <- df_multi[train_data, ]
test_set  <- df_multi[-train_data, ]

# Plotting the tree
tree_two <- predict(prune_two, test_set, type = "class")
mean(tree_two == test_set$marijuana_use_level)
```

```
## [1] 0.8659794
```

```r
# RANDOM FOREST
# Dropping unused factor levels
train_set$marijuana_use_level <- droplevels(train_set$marijuana_use_level)
test_set$marijuana_use_level <- droplevels(test_set$marijuana_use_level)

set.seed(42)
```

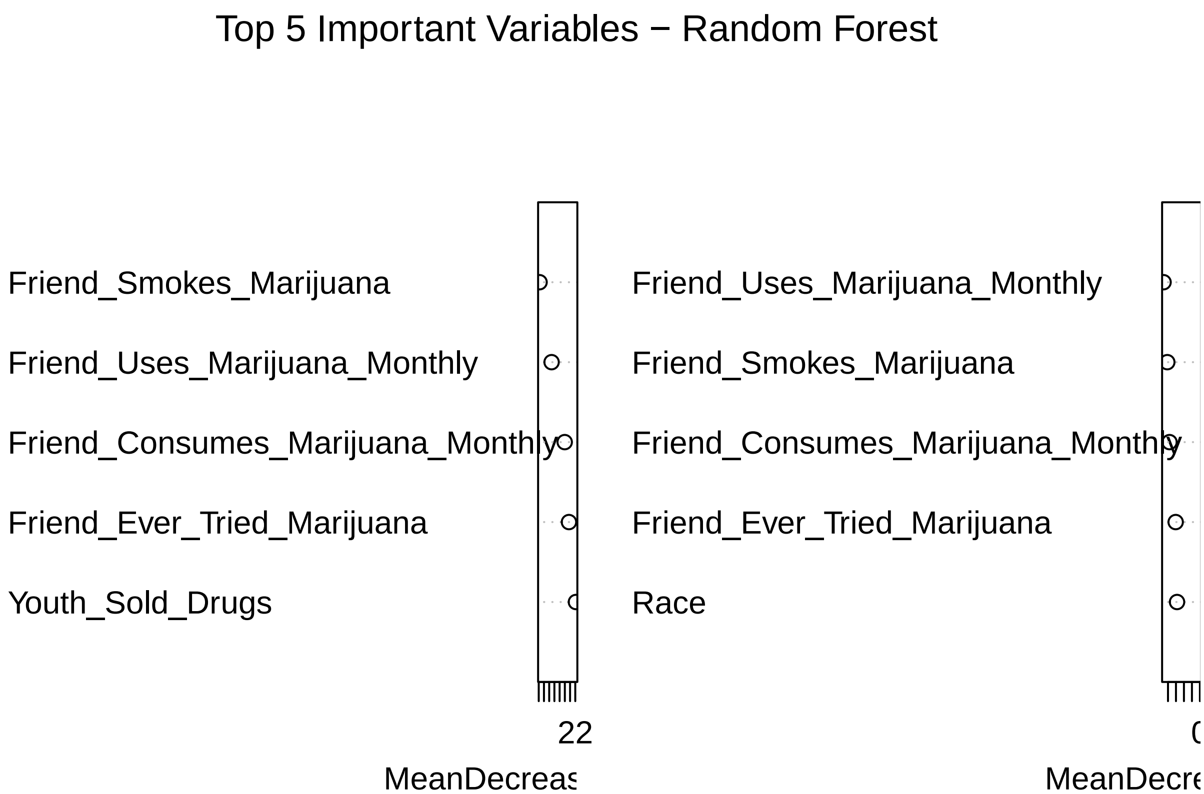
```r
# mtry is set to 5, as we took sqrt(p)
rf_two <- randomForest(marijuana_use_level ~ ., data = train_set, mtry = 5, importance = TRUE, ntree =

prediction_two <- predict(rf_two, test_set, type = "class")
prediction_two <- factor(prediction_two, levels = levels(test_set$marijuana_use_level))
cat("Random Forest Accuracy:", mean(prediction_two == test_set$marijuana_use_level), "\n")
```

```
## Random Forest Accuracy: 0.8932686
```

```r
confusionMatrix(prediction_two, test_set$marijuana_use_level)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction     RARE OCCASIONAL
##   RARE           78         29
##   OCCASIONAL    147       1395
##
##                Accuracy : 0.8933
##                  95% CI : (0.8774, 0.9078)
##     No Information Rate : 0.8636
##     P-Value [Acc > NIR] : 0.0001673
##
##                   Kappa : 0.4188
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.34667
##             Specificity : 0.97963
##          Pos Pred Value : 0.72897
##          Neg Pred Value : 0.90467
##              Prevalence : 0.13645
##          Detection Rate : 0.04730
##    Detection Prevalence : 0.06489
##       Balanced Accuracy : 0.66315
##
##        'Positive' Class : RARE
##
```

```r
varImpPlot(rf_two, n.var = 5, main = "Top 5 Important Variables - Random Forest")
```



In the multi-class classification, the initial decision tree misclassified 12.5 percent of cases, and pruning increased accuracy to 86.6 percent

Random Forest achieved 89.3 percent accuracy with a balanced accuracy of 66.3 percent

Friends consuming marijuana emerged as the most powerful predictor of usage frequency.

```r
# PART3: REGRESSION: number of days per year a person has consumed alcohol
# Performing some data cleaning on IRALCFM, and ignoring 91 values, which is "did not drink", so we set
drug_use$alcohol_days_past_month <- ifelse(
  drug_use$IRALCFM %in% 1:30,
  drug_use$IRALCFM,
  ifelse(drug_use$IRALCFM %in% 91,0,NA)
)

# The dataframe 'df_reg' consists of the predictors and outcome variables for the regression model and
df_reg <- drug_use[, c(demographic_cols, youth_experience_cols, "alcohol_days_past_month")]
df_reg <- na.omit(df_reg)
print(table(df_reg$alcohol_days_past_month))
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   14   15   16
## 6199  247  143   89   43   42    9   14    5    1   14    1    5    2    6    2
##   17   20   21   28   30
##    1    1    1    2    2
```

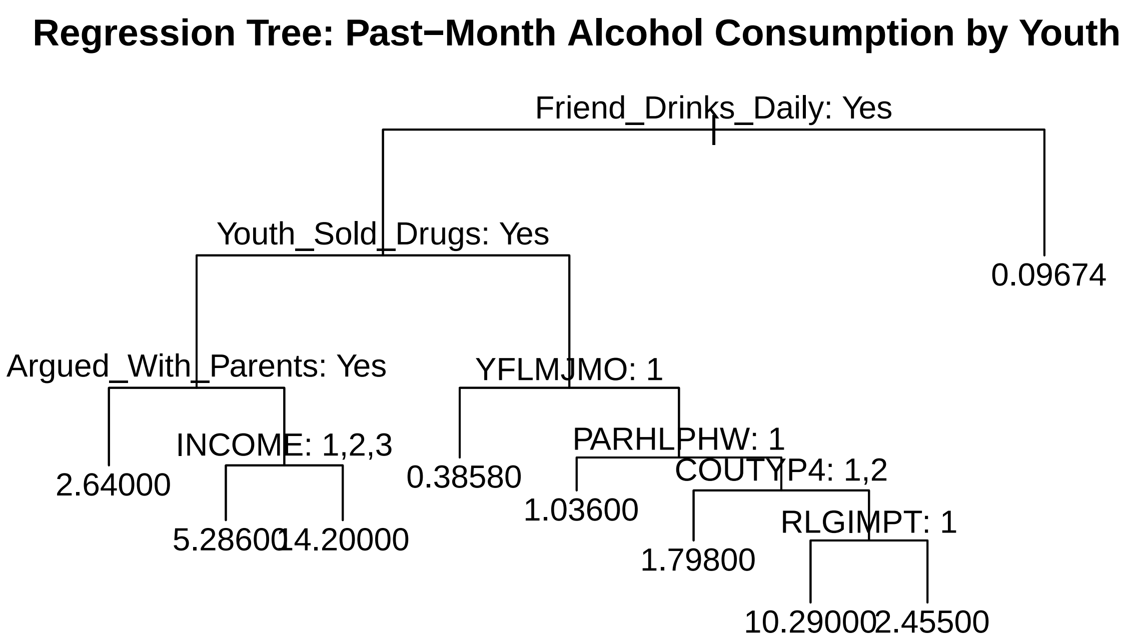
```r
# For readability and clarity of predictor variable names
# Rename variables in df_reg for better readability
colnames(df_reg)[colnames(df_reg) == "FRDMJMON"] <- "Friend_Uses_Marijuana_Monthly"
colnames(df_reg)[colnames(df_reg) == "STNDSMJ"]  <- "Friend_Smokes_Marijuana"
colnames(df_reg)[colnames(df_reg) == "RLGFRND"]  <- "Religious_Friend"
colnames(df_reg)[colnames(df_reg) == "YFLTMRJ2"] <- "Friend_Influence_Marijuana"
colnames(df_reg)[colnames(df_reg) == "FRDMEVR2"] <- "Friend_Ever_Used_Marijuana"
colnames(df_reg)[colnames(df_reg) == "HEALTH2"]  <- "Self_Reported_Health"
colnames(df_reg)[colnames(df_reg) == "YOSELL2"]  <- "Youth_Sold_Drugs"
colnames(df_reg)[colnames(df_reg) == "STNDALC"]  <- "Friend_Drinks_Daily"
colnames(df_reg)[colnames(df_reg) == "EDUSCHGRD2"] <- "Grade_Level"
colnames(df_reg)[colnames(df_reg) == "NEWRACE2"] <- "Race_Category"
colnames(df_reg)[colnames(df_reg) == "ARGUPAR"]  <- "Argued_With_Parents"


# For plot readability, recoding categorical variables.
df_reg$Friend_Drinks_Daily <- factor(df_reg$Friend_Drinks_Daily, levels = c(1, 2), labels = c("Yes", "No

df_reg$Youth_Sold_Drugs <- factor(df_reg$Youth_Sold_Drugs, levels = c(1, 2), labels = c("Yes", "No"))

df_reg$Argued_With_Parents <- factor(df_reg$Argued_With_Parents, levels = c(1, 2), labels = c("Yes", "No


# Splitting the data into train and test data, 70% train data and 30% test data
set.seed(42)
train_index <- sample(1:nrow(df_reg), 0.7 * nrow(df_reg))
train_set <- na.omit(df_reg[train_index, ])
test_set  <- na.omit(df_reg[-train_index, ])

# DECISION TREE
tree_three <- tree(alcohol_days_past_month ~ ., data = train_set)
tree_three
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 4780 10100.00  0.28580
##    2) Friend_Drinks_Daily: Yes 1162  7186.00  0.87440
##      4) Youth_Sold_Drugs: Yes 37  1278.00  4.70300
##        8) Argued_With_Parents: Yes 25   129.80  2.64000 *
##        9) Argued_With_Parents: No 12   820.00  9.00000
##         18) INCOME: 1,2,3 7    87.43  5.28600 *
##         19) INCOME: 4 5   500.80 14.20000 *
##      5) Youth_Sold_Drugs: No 1125  5348.00  0.74840
##       10) YFLMJMO: 1 749  1977.00  0.38580 *
##       11) YFLMJMO: 2 376  3076.00  1.47100
##         22) PARHLPHW: 1 249   888.70  1.03600 *
##         23) PARHLPHW: 2 127  2048.00  2.32300
##           46) COUTYP4: 1,2 109   689.60  1.79800 *
##           47) COUTYP4: 3 18  1146.00  5.50000
##             94) RLGIMPT: 1 7   815.40 10.29000 *
##             95) RLGIMPT: 2 11    68.73  2.45500 *
##    3) Friend_Drinks_Daily: No 3618  2378.00  0.09674 *
```

```r
summary(tree_three)
```

```
##
## Regression tree:
## tree(formula = alcohol_days_past_month ~ ., data = train_set)
## Variables actually used in tree construction:
## [1] "Friend_Drinks_Daily" "Youth_Sold_Drugs"    "Argued_With_Parents"
## [4] "INCOME"              "YFLMJMO"             "PARHLPHW"
## [7] "COUTYP4"             "RLGIMPT"
## Number of terminal nodes:  9
## Residual mean deviance:  1.58 = 7536 / 4771
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -14.20000  -0.09674  -0.09674   0.00000  -0.09674  29.61000
```

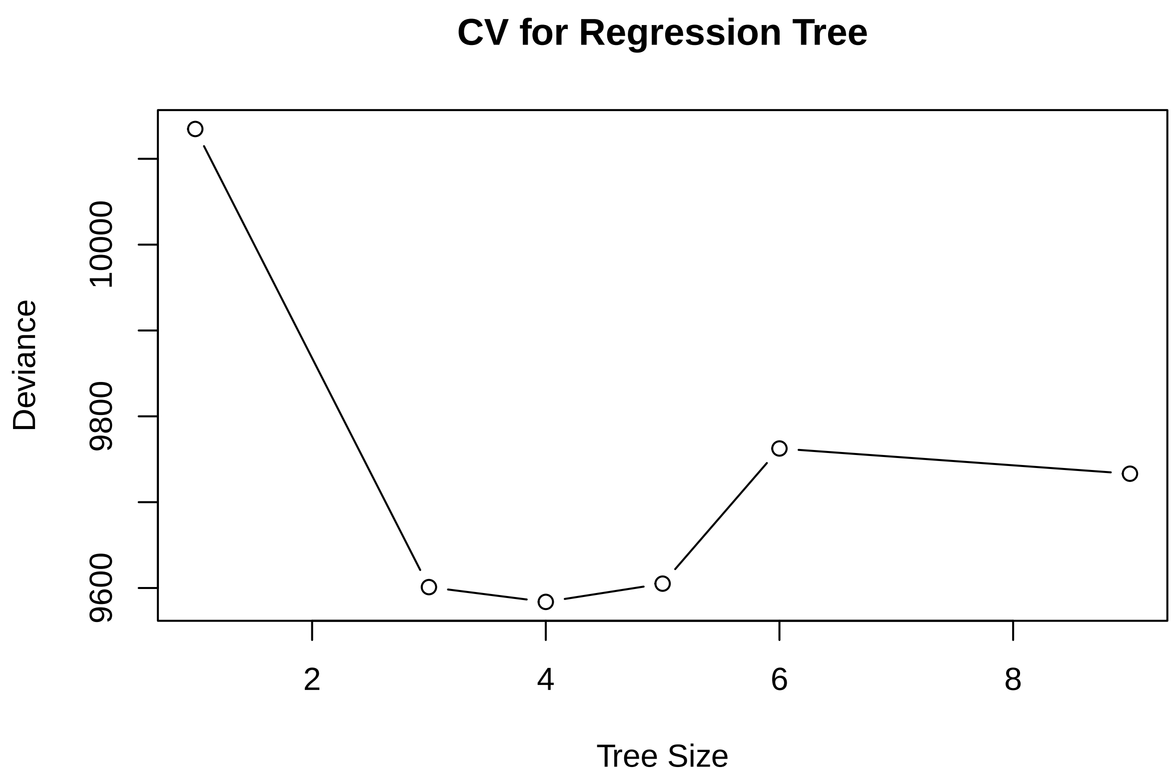
```r
plot(tree_three)
text(tree_three, pretty = 0)
title("Regression Tree: Past-Month Alcohol Consumption by Youth")
```




```r
prediction_three <- predict(tree_three, test_set)
mean((prediction_three - test_set$alcohol_days_past_month)^2)
```

```
## [1] 1.460152
```

```r
# PRUNING THE TREE
set.seed(123)
cv_three <- cv.tree(tree_three)
plot(cv_three$size, cv_three$dev, type = "b", xlab = "Tree Size", ylab = "Deviance", main = "CV for Reg
```



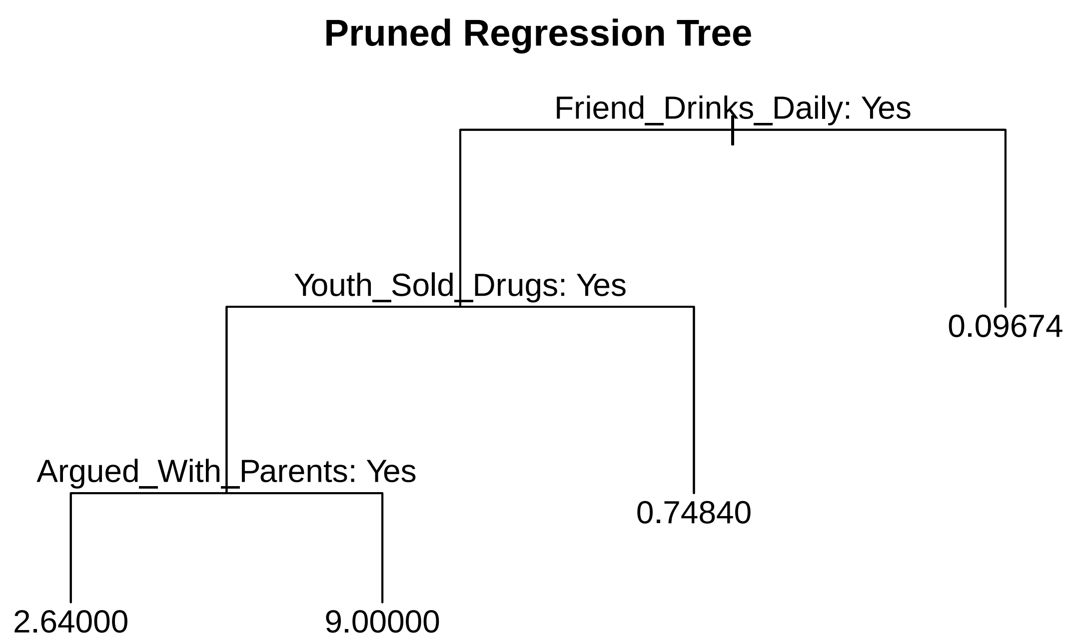

```r
opt_size <- cv_three$size[which.min(cv_three$dev)]
cat("Optimal Tree Size:", opt_size, "\n")
```

```
## Optimal Tree Size: 4
```

```r
# so we take 4 from the graph
prune_three <- prune.tree(tree_three, best = 4)
summary(prune_three)
```

```
##
## Regression tree:
## snip.tree(tree = tree_three, nodes = c(9L, 5L))
## Variables actually used in tree construction:
## [1] "Friend_Drinks_Daily" "Youth_Sold_Drugs"    "Argued_With_Parents"
## Number of terminal nodes:  4
## Residual mean deviance:  1.817 = 8676 / 4776
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -9.00000 -0.09674 -0.09674  0.00000 -0.09674 29.25000
```

```r
plot(prune_three)
text(prune_three, pretty = 0)
title("Pruned Regression Tree")
```




```r
prediction_three <- predict(prune_three, test_set)
mean((prediction_three - test_set$alcohol_days_past_month)^2)
```

```
## [1] 1.428887
```

```r
mse <- mean((prediction_three - test_set$alcohol_days_past_month)^2)
rss <- sum((prediction_three - test_set$alcohol_days_past_month)^2)
tss <- sum((test_set$alcohol_days_past_month - mean(test_set$alcohol_days_past_month))^2)
r_squared <- 1 - rss/tss

cat("MSE is:", mse, "\n")
```

```
## MSE is: 1.428887
```

```r
cat("R squared is:", r_squared, "\n")
```

```
## R squared is: 0.06379351
```

```r
# Note: The outcome is in days
```

```r
# BAGGING
# Using all predictors for bagging
bag_three <- randomForest(alcohol_days_past_month ~ ., data = train_set, mtry = ncol(train_set) - 1, imp
```

```r
prediction_bg <- predict(bag_three, test_set, type = "class")
```

```r
mean((prediction_bg - test_set$alcohol_days_past_month)^2)
```

```
## [1] 1.283587
```

```r
varImpPlot(bag_three, n.var = 5, sort = TRUE, main = "Top 5 Important Variables (Regression)")
```