Exercise06DecisionTrees

Akhila

11/07/2019

What factors explain excessive alcohol consumption among students? The record for the task sheet comes from a survey of students who attended mathematics and Portuguese courses and contains many interesting details about their sociodemographics, life circumstances and learning success.
The ordinal scaled variables `Dalc` and `Walc` give information about the alcohol consumption of the students on weekdays and weekends. Create a binary target variable `alc_prob` as follows:

```
options(repos=structure(c(CRAN="http://cran.r-project.org")))
options(repos="https://cran.rstudio.com" )
install.packages("DescTools")
```

```
## package 'DescTools' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\AKMANI\AppData\Local\Temp\RtmpGEmNOX\downloaded_packages
```

```
install.packages("dineq")
```

```
## package 'dineq' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\AKMANI\AppData\Local\Temp\RtmpGEmNOX\downloaded_packages
```

```
install.packages("party")
```

```
## package 'party' successfully unpacked and MD5 sums checked
##
```

```
## The downloaded binary packages are in
##  C:\Users\AKMANI\AppData\Local\Temp\RtmpGEmNOX\downloaded_packages
```

```
library(stringr)
library(readr)
library(dplyr)
library(DescTools)
library(reldist)
library(party)
library(dineq)
# (Adapt Path)
setwd("C:\\Users\\AKMANI\\Desktop\\DKE_OVGU\\Semester 3\\Visual Analytics\\Exercise\\Exercise 06")
student <- read_csv("student_alc.csv")
student <- student %>%
mutate(alc_prob = ifelse(Dalc + Walc >= 6, "alc_p", "no_alc_p"))
student1<-student
```

1. Calculate the Gini index for the target variable `alc_prob` and the *Gini index* for each variable with respect to `alc_prob`. Determine the 5 variables with the highest *Gini Gain*.

```
# Solution for Task 1
alc_p<-nrow(subset(student, alc_prob=="alc_p"))

no_alc_p<-nrow(subset(student, alc_prob=="no_alc_p"))

tot<-nrow(student)

#calculating GIni of Target variable
gini_tar<- 1-((alc_p/tot)^2)-((no_alc_p/tot)^2)
print("The gini index of targer variable:")
```

```
## [1] "The gini index of targer variable:"
```

```
print(gini_tar)
```

```
## [1] 0.3044897
```

```
#converting all  character columns to factor data type
student[sapply(student, is.character)] <- lapply(student[sapply(student, is.character)],
                                       as.factor)

#converting all  numeric columns to factor data type
student[sapply(student, is.numeric)] <- lapply(student[sapply(student, is.numeric)],
                                       as.factor)

student$absences<-student1$absences
student$G1<-student1$G1
student$G2<-student1$G2
student$G3<-student1$G3

#dividing absences into labels <10, >=10
summary(student$absences)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   4.000   5.709   8.000  75.000
```

```
for(i in 1:nrow(student)){#dividing by range
  if(student$absences[i]  >= 0 && student$absences[i]  <= 10){
    student$absences[i] <- 1
  }
  else if(student$absences[i] > 10){
    student$absences[i] <- 2
  }
}
student$absences<-as.factor(student$absences)

#dividing G1 into labels >= 3 and < 10, >=10 and <=19
summary(student$G1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.00    8.00   11.00   10.91   13.00   19.00
```

```
for(i in 1:nrow(student)){#dividing by range
  if(student$G1[i]  >= 3 && student$G1[i]  <= 10){
    student$G1[i] <- 1
  }
  else if(student$G1[i] > 10 && student$G1[i] <=19){
    student$G1[i] <- 2
  }
}
student$G1<-as.factor(student$G1)

#dividing G2 into labels >= 3 and < 10, >=10 and <=19
summary(student$G2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.00   11.00   10.71   13.00   19.00
```

```
for(i in 1:nrow(student)){#dividing by range
  if(student$G2[i]  >= 0 && student$G2[i]  <= 10){
    student$G2[i] <- 1
  }
  else if(student$G2[i] > 10 && student$G2[i] <=19){
    student$G2[i] <- 2
  }
}
student$G2<-as.factor(student$G2)

summary(student$G3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    8.00   11.00   10.42   14.00   20.00
```

```r
for(i in 1:nrow(student)){#dividing by range
  if(student$G3[i]  >= 0 && student$G3[i]  <= 10){
    student$G3[i] <- 1
  }
  else if(student$G3[i] > 10 && student$G3[i] <=20){
    student$G3[i] <- 2
  }
}
student$G3<-as.factor(student$G3)

df_final<-data.frame(Feat_name=character(), Gini_index=double())

#student <- read_csv("C:/Users/MANAS MANGARAJ/Documents/3rd Semester/Visual Analytics/Exercise6/student_alc.csv")
#student <- student %>%
#mutate(alc_prob = ifelse(Dalc + Walc >= 6, "alc_p", "no_alc_p"))

#str(student)

for(i in colnames(student)){

  df2 <- student %>% group_by(student[[i]],alc_prob) %>% tally()
  names(df2)[1] <- "levl"

  df3<-df2 %>% group_by(levl) %>% summarise(.,n = sum(n))
  names(df3)[2] <- "n1"

  df4<-df2 %>% inner_join(df3)

  df4$prob<- df4$n/df4$n1

  df4$probsq<- df4$prob * df4$prob

  df5<-df4 %>% group_by(levl,n1) %>% summarise(.,probsq = sum(probsq))
  df5$gini<- 1 - df5$probsq

  df5$wtgin<- (df5$n1/nrow(student)) * df5$gini
```

```
  res<-sum(df5$wtgin)

  res1<- gini_tar - res

  nwln<- list(Feat_name= i, Gini_index = res1)
  df_final = rbind(df_final,nwln, stringsAsFactors=FALSE)
  res<-0
  res1<-0
  #break()
}
df<-df_final[order(-df_final$Gini_index),]
head(df,n=10)
```

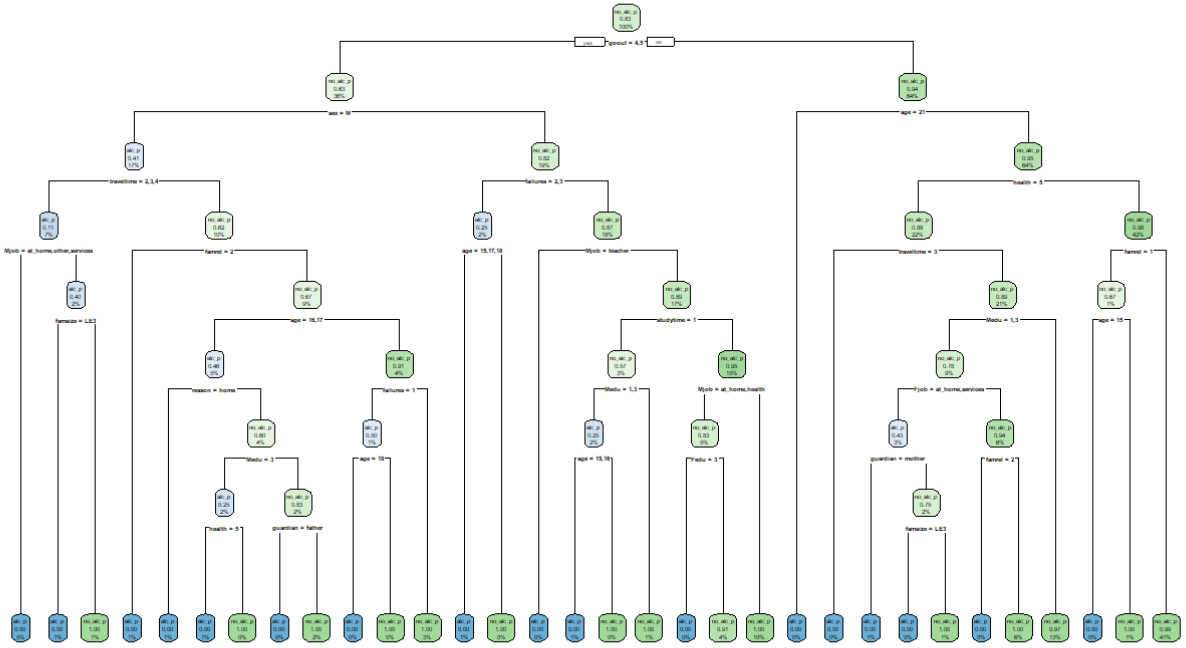
```
##     Feat_name  Gini_index
## 32   alc_prob 0.304489665
## 26       Walc 0.205431586
## 25       Dalc 0.192167805
## 24      goout 0.050476625
## 1         sex 0.027122951
## 12  studytime 0.019114787
## 11 traveltime 0.018197912
## 13   failures 0.013202153
## 23   freetime 0.011198571
## 2         age 0.009302622
```

```
View(df)
```

2. Learn 2 different decision trees with `alc_prob` as target variable. For the first tree, nodes should be further partitioned until the class distribution of all resulting leaf nodes is pure. For the second tree, nodes with a cardinality of less than 20 instances should not be further partitioned. Determine the quality of the trees by calculating sensitivity (*True Positive Rate*) and specificity (*True Negative Rate*) for a 70%:30% split in training and test sets. Display the decision trees graphically and discuss the differences in quality measures

```
# Solution for Task 2

library(rpart)
```

```
library(rpart.plot)
#1st decision tree with pure nodes
stud2<-student
stud2$Walc<-NULL
stud2$Dalc<-NULL
stud2$alc_prob<-factor(student$alc_prob)

#converting character data into factor
stud2 <- mutate_if(stud2, is.character, as.factor)

partdata<-sample(2,nrow(stud2), replace = TRUE, prob = c(0.7,0.3))
train<-stud2[partdata==1,]
test<-stud2[partdata==2,]

#str(train)
tree1<-rpart(alc_prob~., data = train,method = "class", control=rpart.control(minsplit=0))
rpart.plot(tree1, extra = 106)
```



```
print(tree1)
```

```
## n= 262
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##   1) root 262 45 no_alc_p (0.171755725 0.828244275)
##     2) goout=4,5 94 35 no_alc_p (0.372340426 0.627659574)
```

```
##       4) sex=M 44 18 alc_p (0.590909091 0.409090909)
##         8) traveltime=2,3,4 18  2 alc_p (0.888888889 0.111111111)
##          16) Mjob=at_home,other,services 13  0 alc_p (1.000000000 0.000000000) *
##          17) Mjob=health,teacher 5  2 alc_p (0.600000000 0.400000000)
##            34) famsize=LE3 3  0 alc_p (1.000000000 0.000000000) *
##            35) famsize=GT3 2  0 no_alc_p (0.000000000 1.000000000) *
##         9) traveltime=1 26 10 no_alc_p (0.384615385 0.615384615)
##          18) famrel=2 2  0 alc_p (1.000000000 0.000000000) *
##          19) famrel=3,4,5 24  8 no_alc_p (0.333333333 0.666666667)
##            38) age=16,17 13  6 alc_p (0.538461538 0.461538462)
##              76) reason=home 3  0 alc_p (1.000000000 0.000000000) *
##              77) reason=course,other,reputation 10  4 no_alc_p (0.400000000 0.600000000)
##               154) Medu=3 4  1 alc_p (0.750000000 0.250000000)
##                 308) health=5 3  0 alc_p (1.000000000 0.000000000) *
##                 309) health=4 1  0 no_alc_p (0.000000000 1.000000000) *
##               155) Medu=1,4 6  1 no_alc_p (0.166666667 0.833333333)
##                 310) guardian=father 1  0 alc_p (1.000000000 0.000000000) *
##                 311) guardian=mother 5  0 no_alc_p (0.000000000 1.000000000) *
##            39) age=15,18,19 11  1 no_alc_p (0.090909091 0.909090909)
##              78) failures=1 2  1 alc_p (0.500000000 0.500000000)
##               156) age=18 1  0 alc_p (1.000000000 0.000000000) *
##               157) age=19 1  0 no_alc_p (0.000000000 1.000000000) *
##              79) failures=0,3 9  0 no_alc_p (0.000000000 1.000000000) *
##       5) sex=F 50  9 no_alc_p (0.180000000 0.820000000)
##        10) failures=2,3 4  1 alc_p (0.750000000 0.250000000)
##          20) age=15,17,18 3  0 alc_p (1.000000000 0.000000000) *
##          21) age=16 1  0 no_alc_p (0.000000000 1.000000000) *
##        11) failures=0,1 46  6 no_alc_p (0.130434783 0.869565217)
##          22) Mjob=teacher 1  0 alc_p (1.000000000 0.000000000) *
##          23) Mjob=at_home,health,other,services 45  5 no_alc_p (0.111111111 0.888888889)
##            46) studytime=1 7  3 no_alc_p (0.428571429 0.571428571)
##              92) Medu=1,3 4  1 alc_p (0.750000000 0.250000000)
##               184) age=15,16 3  0 alc_p (1.000000000 0.000000000) *
##               185) age=17 1  0 no_alc_p (0.000000000 1.000000000) *
##              93) Medu=4 3  0 no_alc_p (0.000000000 1.000000000) *
##            47) studytime=2,3,4 38  2 no_alc_p (0.052631579 0.947368421)
##              94) Mjob=at_home,health 12  2 no_alc_p (0.166666667 0.833333333)
```

```
##               188) Fedu=3 1  0 alc_p (1.000000000 0.000000000) *
##               189) Fedu=1,2,4 11  1 no_alc_p (0.090909091 0.909090909) *
##              95) Mjob=other,services 26  0 no_alc_p (0.000000000 1.000000000) *
##     3) goout=1,2,3 168 10 no_alc_p (0.059523810 0.940476190)
##       6) age=21 1  0 alc_p (1.000000000 0.000000000) *
##       7) age=15,16,17,18,19,20 167  9 no_alc_p (0.053892216 0.946107784)
##        14) health=5 57  7 no_alc_p (0.122807018 0.877192982)
##          28) traveltime=3 1  0 alc_p (1.000000000 0.000000000) *
##          29) traveltime=1,2 56  6 no_alc_p (0.107142857 0.892857143)
##            58) Medu=1,3 23  5 no_alc_p (0.217391304 0.782608696)
##             116) Fjob=at_home,services 7  3 alc_p (0.571428571 0.428571429)
##               232) guardian=mother 3  0 alc_p (1.000000000 0.000000000) *
##               233) guardian=father,other 4  1 no_alc_p (0.250000000 0.750000000)
##                 466) famsize=LE3 1  0 alc_p (1.000000000 0.000000000) *
##                 467) famsize=GT3 3  0 no_alc_p (0.000000000 1.000000000) *
##             117) Fjob=health,other,teacher 16  1 no_alc_p (0.062500000 0.937500000)
##               234) famrel=2 1  0 alc_p (1.000000000 0.000000000) *
##               235) famrel=1,3,4,5 15  0 no_alc_p (0.000000000 1.000000000) *
##            59) Medu=0,2,4 33  1 no_alc_p (0.030303030 0.969696970) *
##        15) health=1,2,3,4 110  2 no_alc_p (0.018181818 0.981818182)
##          30) famrel=1 3  1 no_alc_p (0.333333333 0.666666667)
##            60) age=15 1  0 alc_p (1.000000000 0.000000000) *
##            61) age=16,18 2  0 no_alc_p (0.000000000 1.000000000) *
##          31) famrel=2,3,4,5 107  1 no_alc_p (0.009345794 0.990654206) *
```

```
testt<-predict(tree1, test, type = 'class')
table_mat <- table(test$alc_prob, testt)
table_mat
```

```
##           testt
##            alc_p no_alc_p
##   alc_p       13       16
##   no_alc_p    17       87
```

```
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
accuracy_Test
```

```
## [1] 0.7518797
```

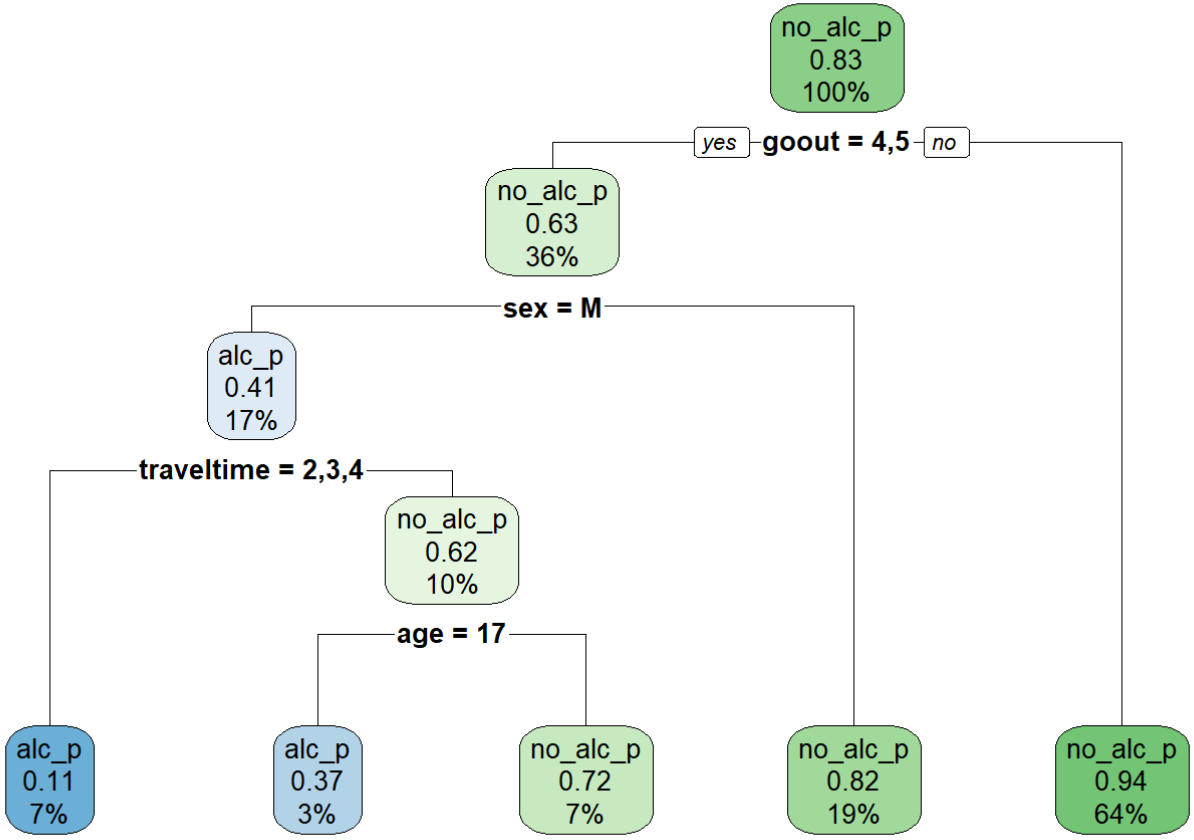
```
TPR<- 8/23
print(TPR)
```

```
## [1] 0.3478261
```

```
TNR<- 12/(12+85)
print(TNR)
```

```
## [1] 0.1237113
```

```
#2nd Decision tree with cardinality 20

tree2<-rpart(alc_prob~., data = train,method = "class", control=rpart.control(minsplit=20))
rpart.plot(tree2, extra = 106)
```




```
testt2<-predict(tree2, test, type = 'class')
table_mat2 <- table(test$alc_prob, testt2)
table_mat2
```

```
##           testt2
##            alc_p no_alc_p
##   alc_p        5       24
##   no_alc_p     8       96
```

```
accuracy_Test2 <- sum(diag(table_mat2)) / sum(table_mat2)
accuracy_Test2
```

```
## [1] 0.7593985
```

```
TPR1<- 4/8
print(TPR1)
```

```
## [1] 0.5
```

```
TNR1<- 16/(16+96)
print(TNR1)
```

```
## [1] 0.1428571
```

```
df_tab1<-data.frame(Name=character(), Accuracy=double(), TPR=double(), TNR=double())

nwln<- list(Name= "1st Tree", Accuracy = accuracy_Test, TPR=TPR, TNR=TNR)
df_tab1 = rbind(df_tab1,nwln, stringsAsFactors=FALSE)

nwln<- list(Name= "2nd Tree", Accuracy = accuracy_Test2, TPR=TPR1, TNR=TNR1)
df_tab1 = rbind(df_tab1,nwln, stringsAsFactors=FALSE)

print(df_tab1)
```

```
##       Name  Accuracy       TPR       TNR
## 1 1st Tree 0.7518797 0.3478261 0.1237113
## 2 2nd Tree 0.7593985 0.5000000 0.1428571
```

```
# Solution for Task 3
```

```r
library(randomForest)

#mtry: Number of variables randomly sampled as candidates at each split.
#ntree: Number of trees to grow.

fit <- randomForest(alc_prob~., data = train, mtry = 5, ntree = 200)
print(fit) # view results
```

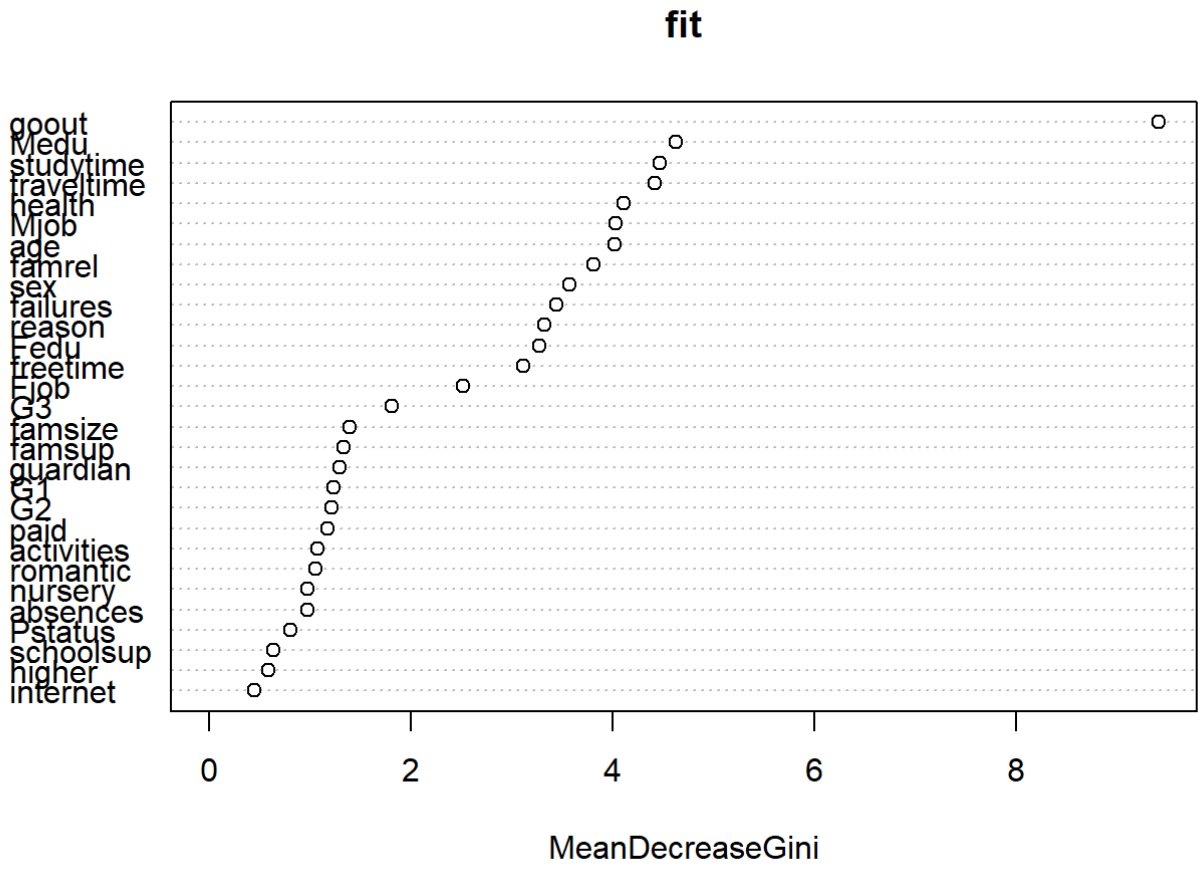
```
##
## Call:
##  randomForest(formula = alc_prob ~ ., data = train, mtry = 5,      ntree = 200)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 14.89%
## Confusion matrix:
##          alc_p no_alc_p class.error
## alc_p        9       36  0.80000000
## no_alc_p     3      214  0.01382488
```

```r
importance(fit) # importance of each predictor
```

```
##            MeanDecreaseGini
## sex               3.5717307
## age               4.0205930
## famsize           1.3910338
## Pstatus           0.8033078
## Medu              4.6308123
## Fedu              3.2762445
## Mjob              4.0273835
## Fjob              2.5182443
## reason            3.3257977
## guardian          1.2945530
## traveltime        4.4209527
```

```
## studytime         4.4650644
## failures          3.4402069
## schoolsup         0.6392798
## famsup            1.3346738
## paid              1.1770884
## activities        1.0687330
## nursery           0.9772091
## higher            0.5820461
## internet          0.4414072
## romantic          1.0550608
## famrel            3.8130576
## freetime          3.1141661
## goout             9.4138971
## health            4.1123640
## absences          0.9694342
## G1                1.2292768
## G2                1.2117783
## G3                1.8103246
```

```
varImpPlot(fit)
```




```
testt3<-predict(fit, test, type = 'class')
table_mat3 <- table(test$alc_prob, testt3)
table_mat3
```

```
##           testt3
##            alc_p no_alc_p
##   alc_p        3       26
##   no_alc_p     2      102
```

```
accuracy_Test3 <- sum(diag(table_mat3)) / sum(table_mat)
accuracy_Test3
```

```
## [1] 0.7894737
```

```
TPR2<- 19/19
print(TPR2)
```

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

```
TNR2<- 96/(98)
print(TNR2)
```

```
## [1] 0.9795918
```

```
df_tab2<-data.frame(Name=character(), Accuracy=double(), TPR=double(), TNR=double())

nwln<- list(Name= "RandomForest Tree", Accuracy = accuracy_Test3, TPR=TPR2, TNR=TNR2)
df_tab2 = rbind(df_tab2,nwln, stringsAsFactors=FALSE)

print(df_tab2)
```

```
##                Name  Accuracy TPR       TNR
## 1 RandomForest Tree 0.7894737   1 0.9795918
```

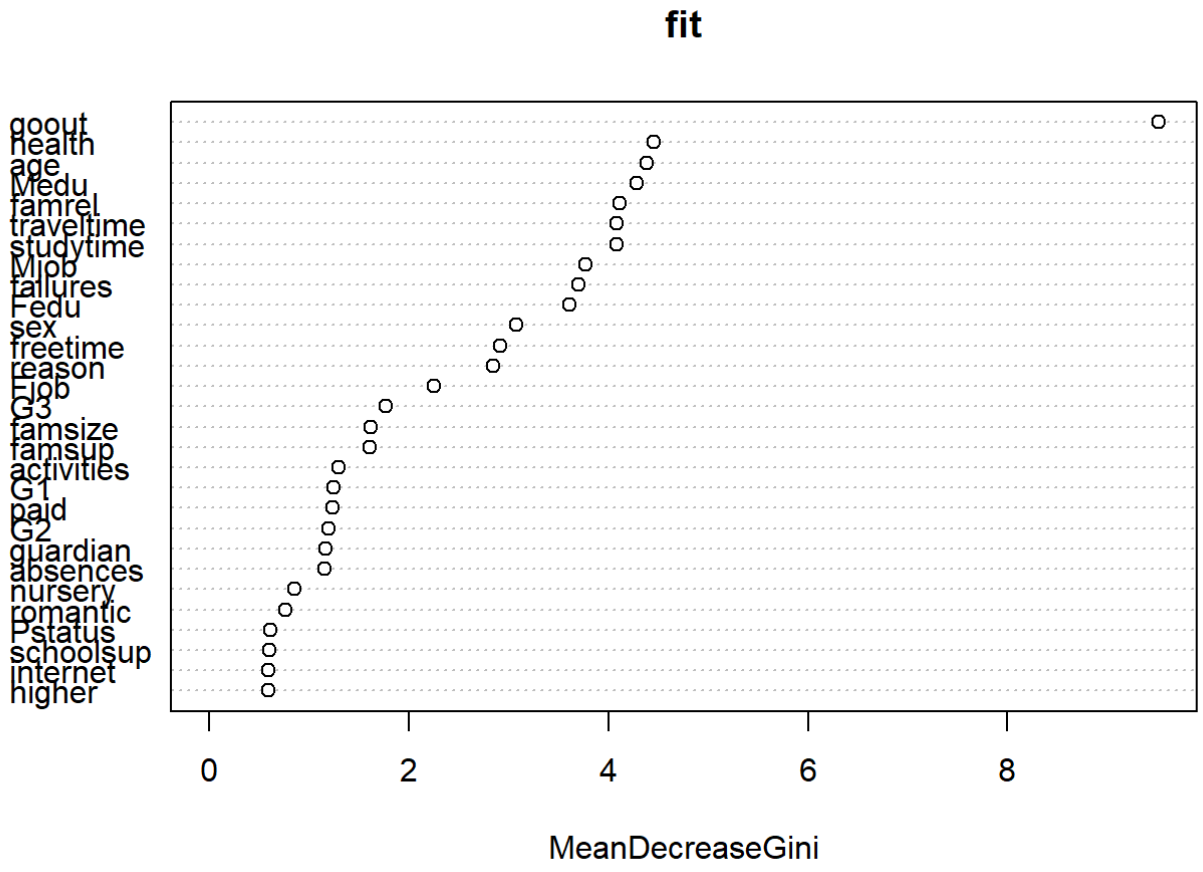
3. Use `randomForest::randomForest()` to create a random forest with 200 trees. As candidates for a split within a tree a random sample of 5 variables should be drawn. Calculate Accuracy, Sensitivity and Specificity for the Out-of-the-Bag instances and show the most important variables (`?importance`).

```
# Solution for Task 3
```

```r
library(randomForest)

#mtry: Number of variables randomly sampled as candidates at each split.
#ntree: Number of trees to grow.

fit <- randomForest(alc_prob~., data = train, mtry = 5, ntree = 200)
print(fit) # view results
```

```
##
## Call:
##  randomForest(formula = alc_prob ~ ., data = train, mtry = 5,      ntree = 200)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 13.74%
## Confusion matrix:
##          alc_p no_alc_p class.error
## alc_p       11       34  0.75555556
## no_alc_p     2      215  0.00921659
```

```r
#importance(fit) # importance of each predictor
varImpPlot(fit)
```




```
testt3<-predict(fit, test, type = 'class')
table_mat3 <- table(test$alc_prob, testt3)
table_mat3
```

```
##           testt3
##            alc_p no_alc_p
##   alc_p        2       27
##   no_alc_p     3      101
```

```
accuracy_Test3 <- sum(diag(table_mat3)) / sum(table_mat)
accuracy_Test3
```

```
## [1] 0.7744361
```

```
TPR2<- 19/19
print(TPR2)
```

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

```
TNR2<- 96/(98)
print(TNR2)
```

```
## [1] 0.9795918
```

```
df_tab2<-data.frame(Name=character(), Accuracy=double(), TPR=double(), TNR=double())

nwln<- list(Name= "RandomForest Tree", Accuracy = accuracy_Test3, TPR=TPR2, TNR=TNR2)
df_tab2 = rbind(df_tab2,nwln, stringsAsFactors=FALSE)

print(df_tab2)
```

```
##               Name  Accuracy TPR       TNR
## 1 RandomForest Tree 0.7744361   1 0.9795918
```

```
#End3
```

Dataset: http://isgwww.cs.uni-magdeburg.de/cv/lehre/VisAnalytics/material/exercise/datasets/student_alc.csv
(Source: https://www.kaggle.com/uciml/student-alcohol-consumption)