NCHS Survey Data Analysis Report

```
install.packages("NHANES")

library(NHANES)

glimpse(NHANES)

library(dplyr)

install.packages("rplot")

library(rpart)

library(rpart.plot)

library(caret)
```

#Solution 1

```
classify <- rpart( SleepTrouble ~ SleepHrsNight + Depressed, data = NHANES, parms = list(split = "gini"))

rpart.plot(classify)

classify

#gives the following as output

n=7768 (2232 observations deleted due to missingness)


node), split, n, loss, yval, (yprob)

    * denotes terminal node


1) root 7768 1969 No (0.7465242 0.2534758)

2) SleepHrsNight>=5.5 6810 1509 No (0.7784141 0.2215859) *

3) SleepHrsNight< 5.5 958  460 No (0.5198330 0.4801670)

6) Depressed=None 680  278 No (0.5911765 0.4088235) *

7) Depressed=Several,Most 278  96 Yes (0.3453237 0.6546763) *
```

#Solution 2

The variables are split based on SleepHrsNight and Depressed using the gini splitting.

The following are the leaves of the classification tree

#terminal node 1

```
SleepHrsNight>=5.5 6810 1509 No (0.7784141 0.2215859)

#terminal node 2

Depressed=None 680  278 No (0.5911765 0.4088235)

#terminal node 3

Depressed=Several,Most 278  96 Yes (0.3453237 0.6546763)
```

When predicting a new observation, if the observation has SleepHrsNight >=6 or SleepHrsNight <6 and Depressed = None

There is no sleep trouble whereas for SleepHrsNight <6 and Depressed = Several,Most there is troublein sleeping

#solution 3

```
NHANES %>%filter(is.na(SleepTrouble) == F) %>%group_by(SleepTrouble)%>%summarise(n = n())%>%mutate(pct = n/sum(n))
```

```
SleepTrouble      n        pct

<fct>           <int>    <dbl>

1 No             5799     0.746

2 Yes             1973     0.254
```

By this we can tell **25.4% have trouble sleeping**

#solution 4

```
set.seed(1234)

index <- sample(2, nrow(NHANES), replace=TRUE, prob=c(0.75, 0.25))

trainData <- NHANES[index==1,]

testData <- NHANES[index==2,]

testData <- testData %>% filter(is.na(SleepTrouble) == F)

trainData <- trainData %>% filter(is.na(SleepTrouble) == F)

training_tree <- rpart( SleepTrouble ~ SleepHrsNight + Depressed, data = trainData, parms = list(split = "gini"))

predicted_tree <- predict(training_tree, newdata = testData, type = "prob")
```

#Cut-point as 0.5

```
confusionMatrix <- table(predicted_tree[,2] >= 0.5,testData$SleepTrouble)

row.names(confusionMatrix) <- c("No","Yes")

confusionMatrix
```

```
      No  Yes
No  1430  427
Yes  26   42
#Calculating specificity and sensitivity for cut-point 0.5
tpr <- confusionMatrix[4]/(confusionMatrix[4] + confusionMatrix[3])
tpr(Sensitivity) = 0.08955224
tnr <- confusionMatrix[1]/(confusionMatrix[1] + confusionMatrix[2])
tnr(Specificity) = 0.9821429
fpr <- 1 - tnr
fpr = 0.01785714
fnr <- 1 - tpr
fnr  = 0.9104478
accuracy_0.5 <- (confusionMatrix[1] + confusionMatrix[4])/sum(confusionMatrix)
accuracy_0.5 = 0.7646753
```

#Cut-point as 0.25

```
confusionMatrix1 <- table(predicted_tree[,2] >= 0.25,testData$SleepTrouble)
row.names(confusionMatrix1) <- c("No","Yes")
confusionMatrix1
       No      Yes
No   1336     363
Yes   120      106
#Calculating specificity and sensitivity for cut-point 0.25
tpr <- confusionMatrix1[4]/(confusionMatrix1[4] + confusionMatrix1[3])
tpr = 0.2260128
tnr <- confusionMatrix1[1]/(confusionMatrix1[1] + confusionMatrix1[2])
tnr = 0.9175824
fpr <- 1 - tnr
fpr = 0.08241758
fnr <- 1 - tpr
fnr  = 0.7739872
accuracy_0.25 <- (confusionMatrix1[1] + confusionMatrix1[4])/sum(confusionMatrix1)
```

```
accuracy_0.25 = 0.7490909
```

Inference:

The overall accuracy in predicting trouble sleeping using only number of hours slept and depression with cut-point as 0.5 is 0.76.

The overall accuracy in predicting trouble sleeping using only number of hours slept and depression with cut-point as 0.25 is 0.75.

All the values changed except fnr which is almost same.

We can tell that if a person has trouble sleeping then the model (with 0.5) will predict this with 0.9 accuracy, and if a person does not have trouble sleeping then the model will predict this with 0.98and 0.77 accuracy respectively.

Similarly, if a person has trouble sleeping then the model (with 0.25) will predict this with 0.22 accuracy, and if a person does not have trouble sleeping then the model will predict this with 0.91 accuracy.

The model (both with o.5 and 0.25 cut-points) has higher accuracy in predicting trouble sleeping compared to no trouble sleeping, this is due to the high false-negative rate of 0.91.

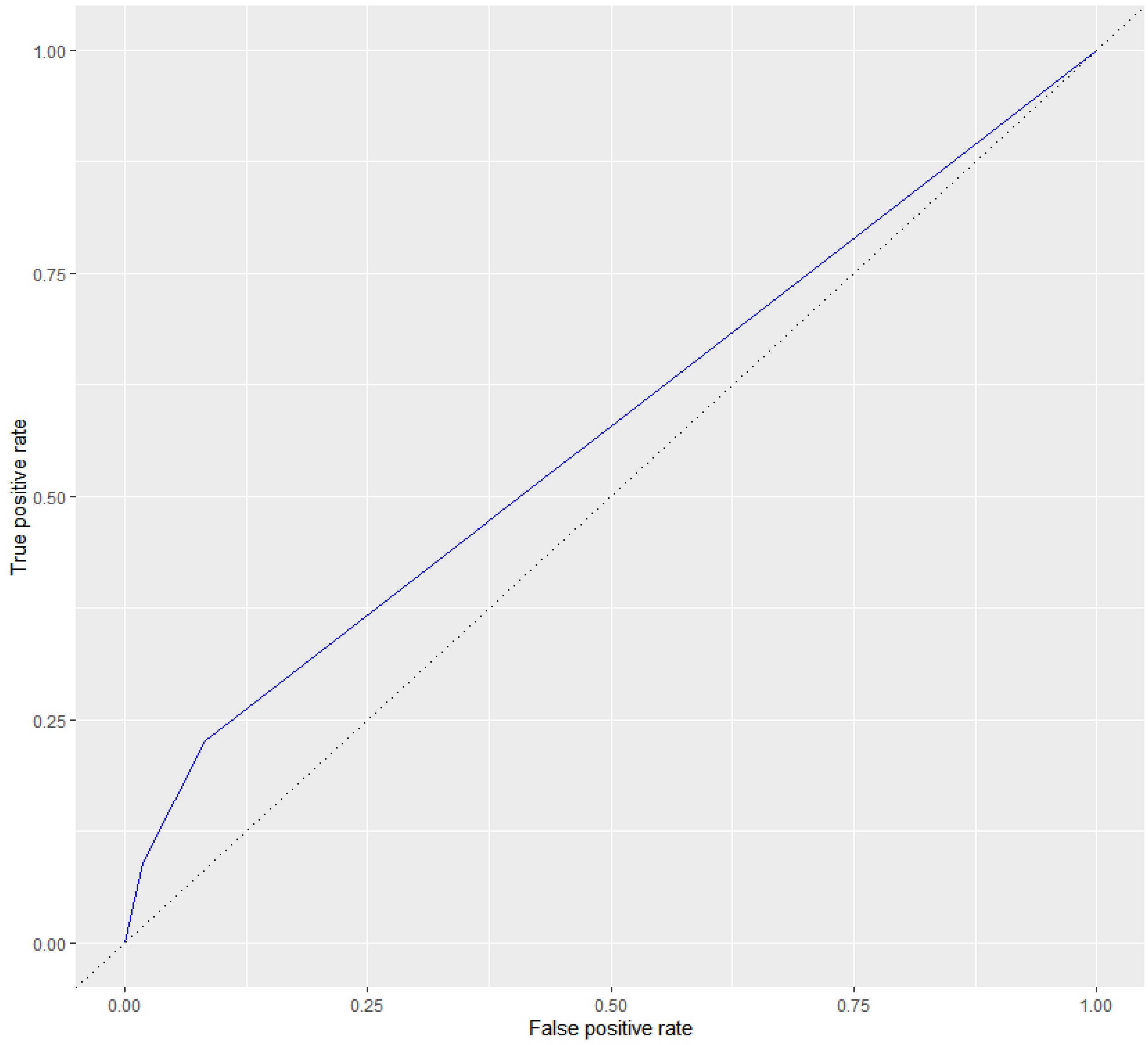
#solution 5

```
require (ROCR)

predicted_tree <- predict (object = classify, newdata = testData, type = "prob")

prediction1 <- prediction (predictions = predicted_tree[,2], testData$SleepTrouble)

performance1 <- performance (prediction1, 'tpr', 'fpr')

perf_df <- data.frame(performance1@x.values, performance1@y.values)

names(perf_df) <- c("fpr", "tpr")

perf_df
```

```
   fpr       tpr
1 0.00000000 0.00000000
2 0.01785714 0.08955224
3 0.08241758 0.22601279
4 1.00000000 1.00000000
```

```
roc <- ggplot(data = perf_df, print.cutoffs.at=seq(0,1,by=0.1), aes(x = fpr, y = tpr)) + geom_line(color = "blue") + geom_abline(intercept = 0, slope = 1, lty = 3) + ylab(performance1@y.name) + xlab(performance1@x.name)
```

Given below is the roc curve



Cut-point 0.5 seems to be the good measure for classifying a person as having sleep trouble, as it has the shorter distance to the corner of the curve

#solution 6

```
tree_full <- rpart(SleepTrouble ~ ., data = trainData, parms = list(split = "gini"))

predicted_tree_full <- predict(object = tree_full, newdata = testData, type = "prob")

confusion_matrix <- table(predicted_tree_full[,2] >= 0.5,testData$SleepTrouble)

row.names(confusion_matrix) <- c("No","Yes")

confusion_matrix
```

```
        No      Yes
```

```
No   1418   404

Yes   38       65

#Accuracy

(confusion_matrix[1] + confusion_matrix[4])/sum(confusion_matrix)

[1] 0.7703896
```

#calculating roc curve using all the variables

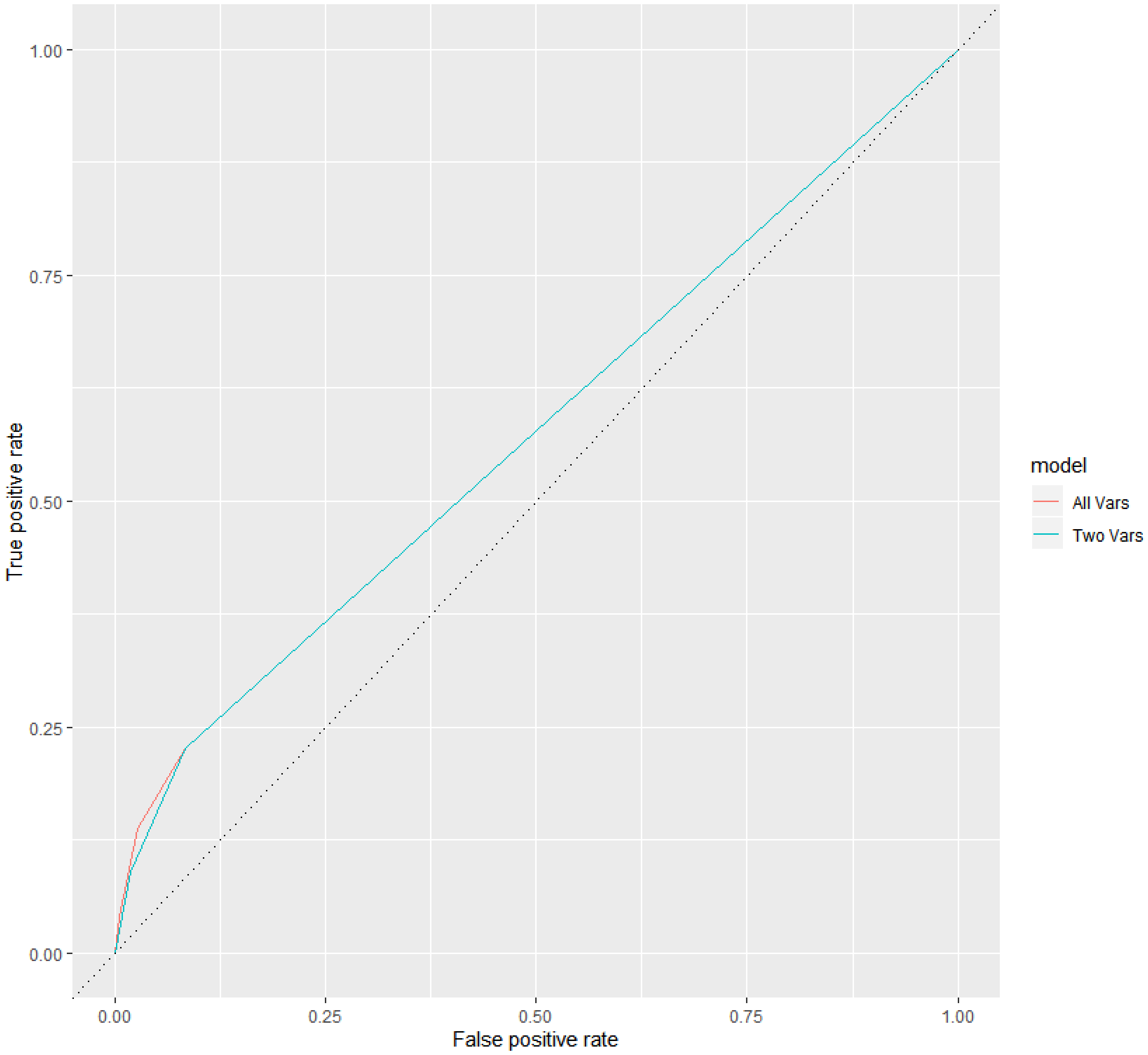
```
pred <- prediction(predictions = predicted_tree_full[,2], testData$SleepTrouble)

perf <- performance(pred, 'tpr', 'fpr')

perf_df_full <- data.frame(perf@x.values, perf@y.values)

names(perf_df_full) <- c("fpr", "tpr")

plot_dat <- cbind(rbind(perf_df_full,perf_df), model = c(rep("All Vars",5),rep("Two Vars",4)))

roc_full <- ggplot(data = plot_dat, aes(x = fpr, y = tpr, colour = model)) + geom_line() +
geom_abline(intercept = 0, slope = 1, lty = 3) +  ylab(perf@y.name) +  xlab(perf@x.name)
```

roc_full is shown in the below picture

Inference:

When used 0.5 as cut-point the accuracy is 76.46% and when taken full variables are taken the accuracy is 77%. We can tell that the classifier with all the variables considered is slightly more accurate compared to the model with cut-out point as 0.5.

This can be seen from the graph as well.


1.00
0.75
0.50
0.25
0.00
True positive rate
0.00
0.25
0.50
0.75
1.00
False positive rate
model
All Vars
Two Vars