The Realation of Stress and Sleep

Liam Amadio, Yanely Ayala, Rolando Castaneda, Lauren Hernandez, Shadae Lewis

Introduction

In today's culture, 'all-nighters' are commonplace and sleep is not taken as seriously as it should be. According to the national sleep foundation 70 to 96 percent of college students get less than eight hours of sleep each night. And over 50 percent of college students sleep less than seven hours per night. A good night's rest can go a really long way in improving health, mood, productivity and stress. There are many factors that go into high quality sleep. There is a combination of many factors that can contribute to stress that our data will explore. Our goal is to understand which predictors correlate with stress level.

To best understand which sleep factors contribute the most to stress, we examine the data provided by Smart-Pillow: An IoT based Device for Stress Detection Considering Sleeping Habits. We analyze sleep data such as the snoring rate, respiration rate, body temperature, limb movement, blood oxygen, eye movement, sleeping hours, and heart rate. In using this sleep data, we are trying to predict which of the aforementioned variables contribute most to the overall stress level of an individual.

About the Data

The data provided was taken not from human test subjects, but an overview received from literature reviews previously published works on sleep and stress. This data set demonstrates the relationship between sleep and stress in a given individual. The utilization of this data is for the purpose of determining what factors are most likely to affect stress levels during sleep. Stress levels are calculated through certain standards met in the other predictors. These testing gradients then rate an individual with a value 1 through 5, 5 being the highest stress level experienced during sleep, and assigned to the stress predictor.

Through analyzing the dataset obtained, we wish to answer the following questions: (1) Are we able to determine someone's stress level based on their body's behavior while sleeping? (2) Which variables are significant when determining the stress level of an individual? In order to determine the response to the above questions, we used Decision trees and Random Forest models. Although recursive binary splitting for decision trees provide good predictions on the training data, it tends to overfit and perform poorly on the test data. To fix this we pruned the tree which selects a subtree that yields the lowest test error rate. Decision trees also struggle with prediction accuracy and suffer from high variance where different subsets of the same data could yield drastically different results. Random Forest is one method to tackle these issues. Random Forests grows B large un-pruned trees but each time a tree split is considered, it picks a random subset of m (sqrt(p) for classification) predictors from the full set of p predictors. This stabilizes the variance of the estimate. We expect the Random Forest test error to be smaller than the test error for a single decision tree.

Variables:

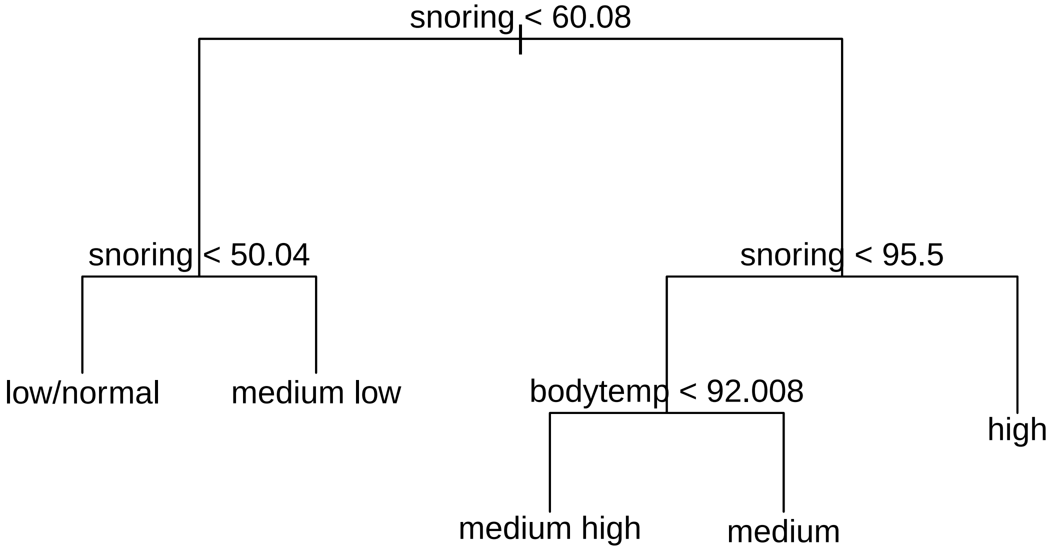
- Column 1: Snoring Rate (snoring)
- Column 2: Respiration Rate (respiration)
- Column 3: Body Temperature (bodytemp)
- Column 4: Limb Movement (limb)

- Column 5: Blood Oxygen (bloodO)
- Column 6: Eye Movement (eye)
- Column 7: Sleeping Hours (sleephrs)
- Column 8: Heart Rate (heart)
- Column 9: Stress Level (stress) – 0-low/normal, 1-medium low, 2-medium, 3-medium high, 4-high

Decision Trees Model

The model for the decision trees is: stress ~ snoring + respiration + bodytemp + limb + bloodO + eye + sleephrs + heart

The values reported in this section were obtained when the seed was set to 2. All variables in the data set were used to create the decision trees because the correlation matrix indicated that all predictors were valuable. First we started with creating a decision tree with the entire data (no training or testing sets). The single tree produced 5 terminal nodes with a misclassification error rate of 0.003175. The data was then separated into training and testing sets using an 80/20 split. An unpruned tree was created which gave the same tree display as the tree created with the entire data set. The unpruned tree gave a training error rate of 0.003968 which is close to the error rate produced from the tree with the entire data. The testing error rate produced was 0.007936508 and the testing accuracy rate was 0.9920635. Since the error rate is close to 0 and the accuracy rate is close to 1, this indicates that the tree is predicting very well.



We next moved on to see if there can be improvements made to the accuracy rate, error rate, and tree display by pruning the tree. The graph from the prune showed that 5 nodes was the best option for the tree. 5 nodes showed a dev of 8 which is significantly smaller than the deviance shown for 4 or less nodes which were 335, 423, 423, and 423 respectively. With both the pruned and unpruned tree having the same number of nodes, this leads to them having the same error rate and accuracy rate. This also means that the decision tree cannot be improved by pruning the data without giving up a significant amount of error.

```
##Cross validation
```

If we were to choose 4 or less nodes even though the deviance is high, this would cause an increase in the error rate and a decrease in the accuracy rate. We do not want to give up too much error in order to have a smaller tree. To prove this we chose to create a tree with 4 nodes. 4 nodes had the second smallest deviance of 335. This tree gave an error rate of 0.2698413 and an accuracy rate of 0.7301587. This error rate is significantly higher than 0.007936508 which is the error rate from the tree with 5 nodes. We are looking at an error percentage of approximately 0% compared to 27%. We do not want to give up 27% of error which is why 5 nodes is best.

```
##Pruned Tree
```

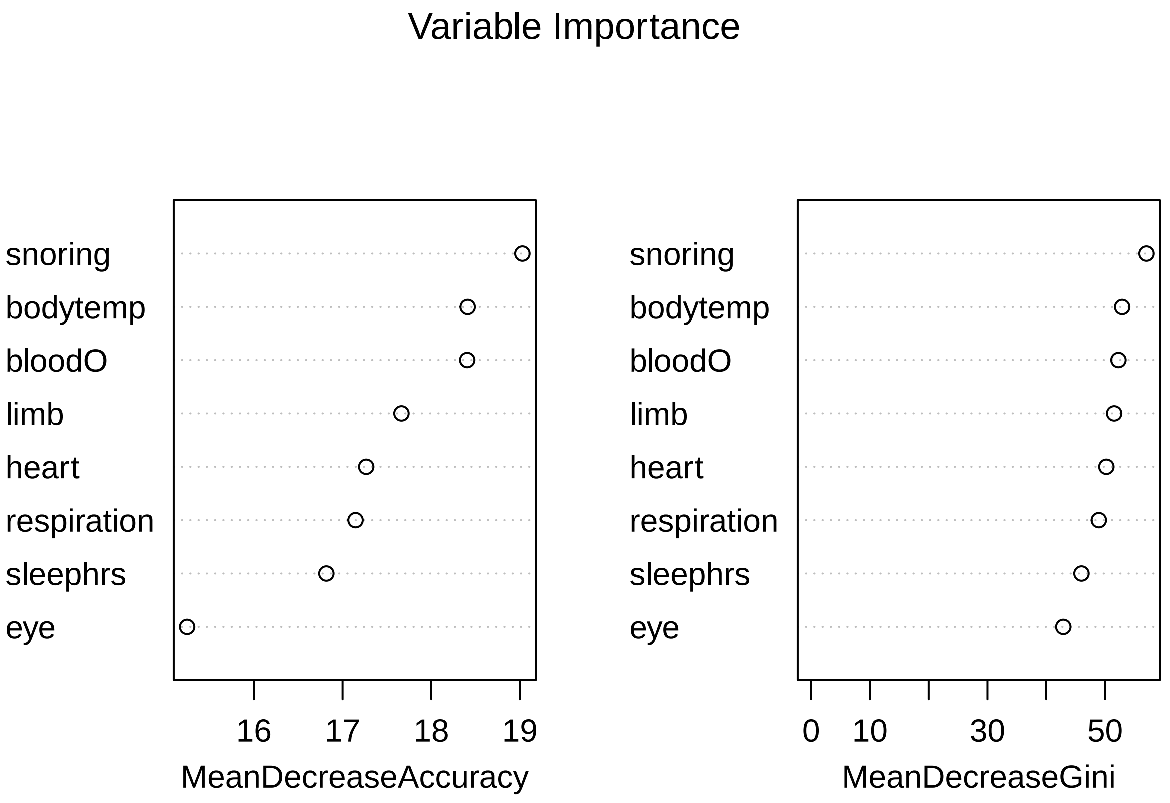
The code was also repeated 10 times with 10 different train/test subdivisions in order to compare error rates.. The seed was set from 1 through 10 with 1 being for the first subdivision and 10 being the last subdivision. The error rates were all close to zero, and the average of all 10 was determined to be 0.01111111. This indicates again that the trees predicted well regardless of what the seed was. Each time the error rate was close to 0 or at zero which indicates high accuracy.

The two predictors displayed in the tree are snoring and bodytemp. This indicates that the two most important variables that are related to stress are the snoring rate and body temperature. There is a higher significance and relationship between snoring, bodytemp and stress versus the other predictors that were not present in the decision tree. More on the importance of each variable will be seen in the random forest section with the use of the importance function.

Random Forest Model

The model formula is:

$stress \sim snoring + respiration + bodytemp + limb + bloodO + eye + sleephrs + heart$



#Conclusion To conclude, through decision trees we found that snoring rate and body temperature are two prevalent predictors in determining the relationship between sleep and stress. Random forests are usually more accurate than decision trees

#Code Appendix

##Importing the data

```
sleep_stress = read.csv(file = "SleepStress.csv",sep=",", header=TRUE,
                        col.names = c("snoring", "respiration", "bodytemp",
                        "limb", "bloodO", "eye","sleephrs", "heart", "stress"))
sleep_stress
```

##Initial Investigations of the data

```
summary(sleep_stress)

#There are no NA's or blank spaces in the data
#The response variable is not binary
```

```
##Correlation
cor(sleep_stress)
pairs(sleep_stress)

#There is a strong relationship between stress levels and all other variables
```

##Decision Trees Model

```
library(tree)
library(boot)
```

```
sleep_stress$stress = as.character(sleep_stress$stress)

##Renaming the values
#(0- low/normal, 1 - medium low, 2- medium, 3-medium high, 4 -high)
sleep_stress$stress[sleep_stress$stress == '0'] = "low/normal"
sleep_stress$stress[sleep_stress$stress == '1'] = "medium low"
sleep_stress$stress[sleep_stress$stress == '2'] = "medium"
sleep_stress$stress[sleep_stress$stress == '3'] = "medium high"
sleep_stress$stress[sleep_stress$stress == '4'] = "high"
```

```
##Before training and testing sets
set.seed(2)

sleep_stress$stress = as.factor(sleep_stress$stress)

tree.sleep.stress = tree(stress~.,sleep_stress)
summary(tree.sleep.stress)

plot(tree.sleep.stress)
text(tree.sleep.stress,pretty = 0)
```

```r
##With training and testing sets

means=c();
for(i in c(1:10)){

set.seed(i)
sample =sample(nrow(sleep_stress), nrow(sleep_stress)*.8)
train = sleep_stress[sample,]
test = sleep_stress[-sample,]

##Unpruned tree
tree.model = tree(stress ~ ., data = sleep_stress,subset = sample)
#summary(tree.model)
#tree.model

##Plotting the unpruned model
#plot(tree.model)
#text(tree.model, pretty=0)

##testing the unpruned model
model.pred = predict(tree.model,test,type = "class")
matrix = table(model.pred,test$stress)
#matrix

##Error rate
error1 = matrix[4,3]/sum(matrix)
#error1

##Accuracy rate
accuracy1 = 1-error1
#accuracy1

##Cross validation
cv.sleep_stress = cv.tree(tree.model,FUN = prune.misclass)
#cv.sleep_stress
#plot(cv.sleep_stress$size,cv.sleep_stress$dev,type = "b")


##Pruned Tree
prune.sleep_stress = prune.misclass(tree.model,best = 5)

##Plotting the pruned model
#plot(prune.sleep_stress)
#text(prune.sleep_stress,pretty = 0)

##Testing the pruned model
tree.pred = predict(prune.sleep_stress,test,type = "class")
matrix2 = table(tree.pred,test$stress)
#matrix2

##Error rate
error2 = matrix2[4,3]/sum(matrix2)
#error2

##Accuracy rate
accuracy2 = 1 - error2
#accuracy2

##Error rates for 10 different subdivisions
means[i]=mean(model.pred != test$stress)
print(means[i])
}

##Average of means
print(mean(means))
```

```
##Prune experiment with 4 nodes
set.seed(2)
prune.check = prune.misclass(tree.model,best = 4)

#Plot
plot(prune.check)
text(prune.check,pretty = 0)

##Testing the pruned model
tree.check = predict(prune.check,test,type = "class")
matrix3 = table(tree.check,test$stress)
matrix3

##Error rate
error3 = (matrix3[4,3] + matrix3[2,5])/sum(matrix3)
error3

##Accuracy rate
accuracy3 = 1 - error3
accuracy3
```

##Random Forest Model

```
library(randomForest)

means.rf = c(); #creating empty list to collect each mean value

for (i in c(1:10))   #setting up the for loop for each seed i
```

```
{
set.seed(i)

sleep_stress$stress = as.factor(sleep_stress$stress)   # setting the data response as a factor
train.rf = sample(nrow(sleep_stress), nrow(sleep_stress)*.8)   # 80/20 training and testing se
test.rf = sleep_stress[-train.rf,]                             # testing set

sleep.rf = randomForest(stress ~., data = sleep_stress,       # random forest model w/ trainin
                        subset = train.rf,
                        mtry=sqrt((ncol(sleep_stress)-1)) ,
                        importance = TRUE)

sleep.rf

yhat.rf = predict(sleep.rf, newdata = test.rf)              # running model w/ test data

#mean((yhat.rf - test.rf$stress)^2)

means.rf[i] = mean((yhat.rf != test.rf$stress)^2)            #




}

print("Mean values of each i:")
print(means.rf)

print(mean(means.rf))

importance(sleep.rf)
varImpPlot(sleep.rf, sort = TRUE, main = "Output")
```

#References

Rachakonda, Laavanya. “Human Stress Detection in and through Sleep.” Kaggle, 15 Feb. 2022, https://www.kaggle.com/datasets/laavanya/human-stress-detection-in-and-through-sleep?select=SaYoPillow.csv.

#Git Repository

https://github.com/cnoott/datascience-project