Duncan McKinnon

West

HW 9

```
# Read in packages and data
suppressPackageStartupMessages({
  library(tidyverse)
  library(NHANES)
  library(neuralnet)
})
```

1).

```
# Create indicator variable RM1 and Sex1
data <- NHANES %>%
  mutate(RM1 = ifelse(RegularMarij=="Yes",1,0), Sex1 = ifelse(Gender=="male",1,0))

# Summarize indicators
data %>%
  select(RM1, Sex1) %>%
  summary()
```

```
##       RM1             Sex1
##  Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:0.000
##  Median :0.000   Median :0.000
##  Mean   :0.276   Mean   :0.498
##  3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :1.000   Max.   :1.000
##  NA's   :5059
```

2).

```
# Limit dataset to HardDrugs, RM1, Age, BMI and Sex1
# Scale Age and BMI to be between 0-1
data <- data %>%
  select(HardDrugs, RM1, Age, BMI, Sex1) %>%
  mutate_at(vars(Age, BMI), funs(scale(.) %>% as.vector))

data %>% glimpse()
```

```
## Observations: 10,000
## Variables: 5
## $ HardDrugs <fct> Yes, Yes, Yes, NA, Yes, NA, NA, No, No, No, No, Yes,...
## $ RM1       <dbl> 0, 0, 0, NA, 0, NA, NA, 0, 0, 0, NA, 1, 1, NA, NA, 0...
## $ Age       <dbl> -0.1224285, -0.1224285, -0.1224285, -1.4618598, 0.54...
## $ BMI       <dbl> 0.753718485, 0.753718485, 0.753718485, -1.540027677,...
## $ Sex1      <dbl> 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1...
```

3).

```
# remove missing values
data <- data %>% na.omit()

# get length of remaining columns
data %>% lapply(length)
```

```
## $HardDrugs
## [1] 4907
##
## $RM1
## [1] 4907
##
## $Age
## [1] 4907
##
## $BMI
## [1] 4907
##
## $Sex1
## [1] 4907
```

4).

```
# Set random seed
set.seed(1847)
p <- 0.2
m <- 4907

# get random indices and split train and test data
train_ind <- sample.int(m, (1-p)*m)
traind <- data[train_ind,]
testd <- data[-train_ind,]

traind %>% glimpse()
```

```
## Observations: 3,925
## Variables: 5
## $ HardDrugs <fct> No, No, No, Yes, No, No, No, No, Yes, No, No, No, No...
## $ RM1       <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0...
## $ Age       <dbl> 0.90446882, 0.41334403, -0.56890557, -0.47961016, -0...
## $ BMI       <dbl> -0.3687530, -0.5504091, 1.7270694, 0.3849839, -0.143...
## $ Sex1      <dbl> 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0...
```

```
testd %>% glimpse()
```

```
## Observations: 982
## Variables: 5
## $ HardDrugs <fct> Yes, No, No, Yes, No, No, No, No, Yes, Yes, No, No, ...
## $ RM1       <dbl> 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0...
## $ Age       <dbl> -0.12242848, 0.36869632, 0.36869632, 0.77052570, 0.0...
## $ BMI       <dbl> 0.75371848, 0.07860880, 0.07860880, -0.08542388, 1.2...
## $ Sex1      <dbl> 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0...
```

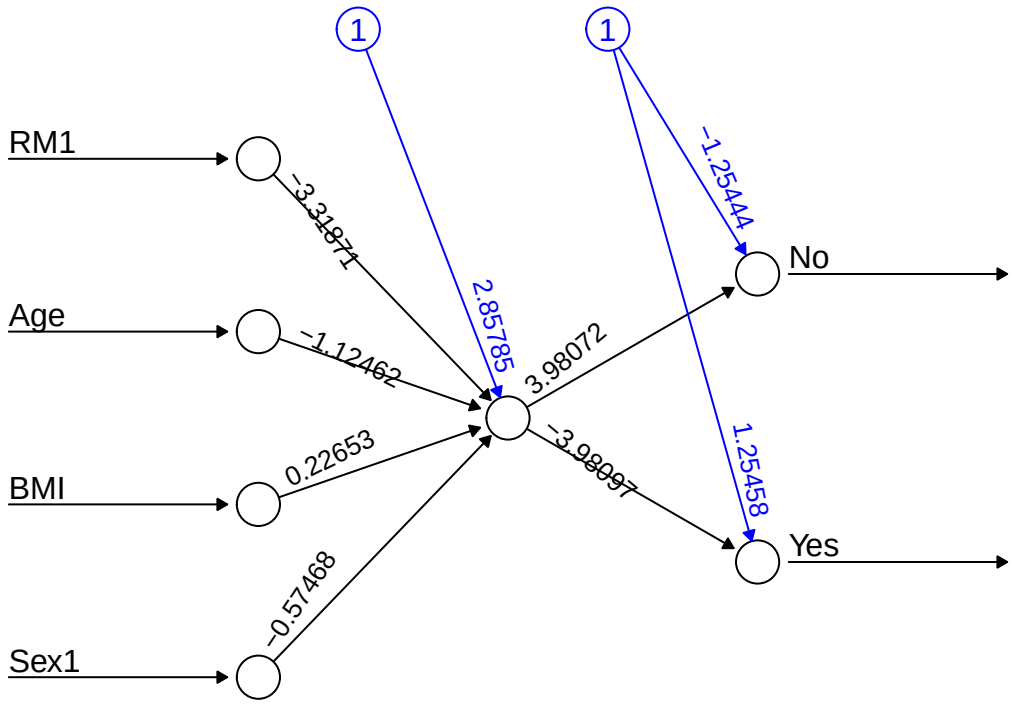
5).

```
# Set random seed
set.seed(1847)

# create formula
f <- formula(HardDrugs ~ RM1 + Age + BMI + Sex1)

# fit the NN on the training data
mNN <- neuralnet(f, traind, hidden=1, linear.output=F, rep=10, lifesign="minimal")
```

```
## hidden: 1    thresh: 0.01    rep:  1/10    steps:    2657    error: 471.62653    time: 2.33 secs
## hidden: 1    thresh: 0.01    rep:  2/10    steps:    1434    error: 471.62676    time: 1.45 secs
## hidden: 1    thresh: 0.01    rep:  3/10    steps:    1166    error: 471.62645    time: 1.17 secs
## hidden: 1    thresh: 0.01    rep:  4/10    steps:     567    error: 471.62666    time: 0.58 secs
## hidden: 1    thresh: 0.01    rep:  5/10    steps:    1469    error: 471.62667    time: 1.5 secs
## hidden: 1    thresh: 0.01    rep:  6/10    steps:    1481    error: 471.62639    time: 1.55 secs
## hidden: 1    thresh: 0.01    rep:  7/10    steps:     441    error: 471.6256     time: 0.47 secs
## hidden: 1    thresh: 0.01    rep:  8/10    steps:     909    error: 471.62642    time: 0.92 secs
## hidden: 1    thresh: 0.01    rep:  9/10    steps:     602    error: 471.62635    time: 0.65 secs
## hidden: 1    thresh: 0.01    rep: 10/10    steps:    2242    error: 471.62608    time: 2.28 secs
```

6).

```
# plot network form for result
plot(mNN, rep="best")
```



Error: 471.6256 Steps: 441

7).

```
# get predictions on test data for the best network result
p1 <- compute(mNN, testd, rep=7)

# create confusion matrix
table(round(p1$net.result[,2]), ifelse(testd$HardDrugs=="Yes",1,0))
```

```
##
##       0   1
##   0 717 113
##   1  59  93
```

8).

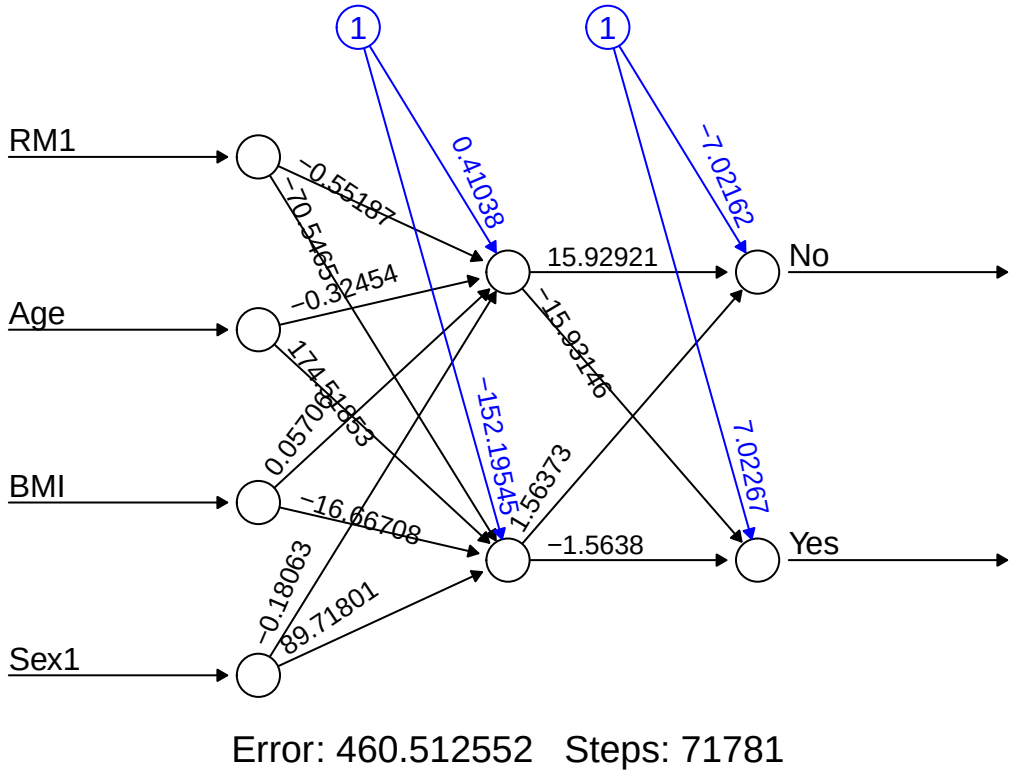
```
# set random seed
set.seed(1847)

# fit the NN on the training data with 2 hidden layers
mNN2 <- neuralnet(f, traind, hidden=2, linear.output=F, rep=10, lifesign="minimal")
```

```
## hidden: 2    thresh: 0.01    rep:  1/10    steps:   16242    error: 462.56932    time: 21.83 secs
## hidden: 2    thresh: 0.01    rep:  2/10    steps:    3332    error: 467.32647    time: 4.37 secs
## hidden: 2    thresh: 0.01    rep:  3/10    steps:   16335    error: 462.54016    time: 22.07 secs
```

```
## hidden: 2    thresh: 0.01    rep:  4/10    steps:   71781    error: 460.51255    time: 1.65 mins
## hidden: 2    thresh: 0.01    rep:  5/10    steps:   17179    error: 462.54186    time: 22.9 secs
## hidden: 2    thresh: 0.01    rep:  6/10    steps:    4714    error: 465.14661    time: 6.18 secs
## hidden: 2    thresh: 0.01    rep:  7/10    steps:    4571    error: 465.05289    time: 5.93 secs
## hidden: 2    thresh: 0.01    rep:  8/10    steps:   12005    error: 467.35928    time: 15.65 secs
## hidden: 2    thresh: 0.01    rep:  9/10    steps:    1257    error: 471.63031    time: 1.63 secs
## hidden: 2    thresh: 0.01    rep: 10/10    steps:    5176    error: 467.32646    time: 6.9 secs
```

9).

```
# plot network form for result
plot(mNN2, rep="best")
```



10).

```
# get predictions on test data for the best network result
p2 <- compute(mNN2, testd, rep=4)

# create confusion matrix
table(round(p2$net.result[,2]), ifelse(testd$HardDrugs=="Yes",1,0))
```

```
##
##       0   1
##   0 720 126
```

```
##   1  56  80
```