Data Wrangling

KO

11/29/2018

```
library(readr)
library(dplyr)

data <- read_delim('revent.TXT', delim = ";")
# head(data)
# ncol(data)

# data <- sapply(data[18:203], function(x) as.numeric(as.character(x)))

dataFinal <- data %>%

  # Filtering for PGA TOUR Stroke Play Events

  filter(`Official Event(Y/N)` == "Y") %>%

  # Changing variables to the proper types

  mutate(`Total Rounds` = as.numeric(`Total Rounds)`),
         `Finish Position(numeric)` = as.numeric(`Finish Position(numeric)`),
         `Birdies` = as.numeric(`Birdies`),
         `Total Holes Over Par` = as.numeric(`Total Holes Over Par`),
         `Drives Over 300 Yards (# of Drives)` = as.numeric(`Drives Over 300 Yards (# of Drives)`),
         `3-Putt Avoid(Total 3 Putts)` = as.numeric(`3-Putt Avoid(Total 3 Putts)`),
         `Avg Distance of Putts Made(Total Distance of Putts)` = as.numeric(`Avg Distance of Putts Made
         `Total Holes Played` = as.numeric(`Total Holes Played`),
         `Total Greens in Regulation` = as.numeric(`Total Greens in Regulation`),
         `App. 50-125 Yards(ft)` = as.numeric(`App. 50-125 Yards(ft)`),
         `App.  50-125 Yards(attempts)` = as.numeric(`App.  50-125 Yards(attempts)`),

         #Creating our desired variables

         cutMade = as.factor(ifelse(`Finish Position(numeric)` < 999, 1, 0)),
         birdiesPerRound = `Birdies` / `Total Rounds`,
         GIRsPerRound = `Total Greens in Regulation` / `Total Holes Played`,
         overParHolesPerRound = `Total Holes Over Par` / `Total Rounds`,
         ThreePuttsPerRound = `3-Putt Avoid(Total 3 Putts)` / `Total Rounds`,
         over300DrivesPerRound = `Drives Over 300 Yards (# of Drives)` /
           `Total Rounds`,
         distPuttsMadePerRound = `Avg Distance of Putts Made(Total Distance of Putts)` / `Total Rounds`
         proxToHoleApproach = `App. 50-125 Yards(ft)` / `App.  50-125 Yards(attempts)`)  %>%

  #Selecting our desired columns

  select(`Player Name`,
        `Event Name`,
        cutMade,
        birdiesPerRound,
```

```
        GIRsPerRound,
        overParHolesPerRound,
        ThreePuttsPerRound,
        over300DrivesPerRound,
        distPuttsMadePerRound,
        proxToHoleApproach)

# https://stackoverflow.com/questions/4862178/remove-rows-with-all-or-some-nas-missing-values-in-data-f
dataFinal <- dataFinal[complete.cases(dataFinal), ]

head(dataFinal)
```

```
## # A tibble: 6 x 10
##   `Player Name`   `Event Name` cutMade birdiesPerRound GIRsPerRound
##   <chr>           <chr>        <fct>             <dbl>        <dbl>
## 1 Allan, Steve    Safeway Open 0                  1.50        0.722
## 2 Ancer, Abraham  Safeway Open 1                  3.25        0.653
## 3 Armour, Ryan    Safeway Open 0                  3.00        0.694
## 4 Atkins, Matt    Safeway Open 0                  3.00        0.500
## 5 Axley, Eric     Safeway Open 0                  2.50        0.556
## 6 Baddeley, Aaron Safeway Open 0                  3.50        0.639
## # ... with 5 more variables: overParHolesPerRound <dbl>,
## #   ThreePuttsPerRound <dbl>, over300DrivesPerRound <dbl>,
## #   distPuttsMadePerRound <dbl>, proxToHoleApproach <dbl>
```

```
#View(dataFinal)
```

```
# Random Forest

library(rpart)
library(partykit)
```

```
## Loading required package: grid
```

```
## Loading required package: libcoin
```

```
## Loading required package: mvtnorm
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
```

```
##     cov, smooth, var
```

```
# ?randomForest

###############
# split 80/20 --------------------------
set.seed(123)

n <- nrow(dataFinal)
train_id <- sample(1:n, size=round(n*0.8)) # select approx 80% of the row numbers between 1 and n
train1 <- dataFinal[train_id,] # the data set we'll train the model on
test1 <- dataFinal[-train_id,] # the data set we'll test the model on

head(train1)
```

```
## # A tibble: 6 x 10
##   `Player Name`   `Event Name`        cutMade birdiesPerRound GIRsPerRound
##   <chr>           <chr>               <fct>             <dbl>        <dbl>
## 1 Gooch, Talor    Genesis Open        1                  3.75        0.569
## 2 Stallings, Sco~ John Deere Classic  0                  2.00        0.806
## 3 Byrd, Jonathan  Houston Open        1                  3.00        0.741
## 4 Kaymer, Martin  PGA Championship    1                  3.00        0.736
## 5 Grillo, Emilia~ THE NORTHERN TRUST  1                  3.25        0.653
## 6 Malnati, Peter  Sanderson Farms Ch~ 1                  4.00        0.694
## # ... with 5 more variables: overParHolesPerRound <dbl>,
## #   ThreePuttsPerRound <dbl>, over300DrivesPerRound <dbl>,
## #   distPuttsMadePerRound <dbl>, proxToHoleApproach <dbl>
```

```
head(test1)
```

```
## # A tibble: 6 x 10
##   `Player Name`   `Event Name` cutMade birdiesPerRound GIRsPerRound
##   <chr>           <chr>        <fct>             <dbl>        <dbl>
## 1 Allan, Steve    Safeway Open 0                  1.50        0.722
## 2 Atkins, Matt    Safeway Open 0                  3.00        0.500
## 3 Baddeley, Aaron Safeway Open 0                  3.50        0.639
## 4 Blixt, Jonas    Safeway Open 1                  3.50        0.597
## 5 Conners, Corey  Safeway Open 1                  3.25        0.736
## 6 Dahmen, Joel    Safeway Open 0                  4.50        0.694
## # ... with 5 more variables: overParHolesPerRound <dbl>,
## #   ThreePuttsPerRound <dbl>, over300DrivesPerRound <dbl>,
## #   distPuttsMadePerRound <dbl>, proxToHoleApproach <dbl>
```

```
# Setting formula for random forest
f <- as.formula(cutMade ~ birdiesPerRound + GIRsPerRound + overParHolesPerRound + ThreePuttsPerRound + o

# Training forest
mod_forest <- randomForest(f, data = train1, ntree = 128, mtry = 2)
mod_forest
```

```
##
## Call:
##  randomForest(formula = f, data = train1, ntree = 128, mtry = 2)
##                Type of random forest: classification
##                      Number of trees: 128
## No. of variables tried at each split: 2
##
```

```
##         OOB estimate of  error rate: 10.43%
## Confusion matrix:
##      0    1 class.error
## 0 1456  179   0.1094801
## 1  220 1970   0.1004566
```

```
sum(diag(mod_forest$confusion)) / nrow(train1)
```

```
## [1] 0.8956863
```

```
# Get importance, code from textbook
library(tibble)
importance(mod_forest) %>%
  as.data.frame() %>%
  rownames_to_column() %>%
  arrange(desc(MeanDecreaseGini))
```

```
##                 rowname MeanDecreaseGini
## 1  overParHolesPerRound         556.1093
## 2       birdiesPerRound         399.7547
## 3 distPuttsMadePerRound         237.9638
## 4    ThreePuttsPerRound         188.8303
## 5          GIRsPerRound         186.4707
## 6    proxToHoleApproach         156.8493
## 7 over300DrivesPerRound         136.7493
```

```
# https://www.r-bloggers.com/how-to-implement-random-forests-in-r/
predTest <- predict(mod_forest, test1, type = "class")

mean(predTest == test1$cutMade)
```

```
## [1] 0.8912134
```

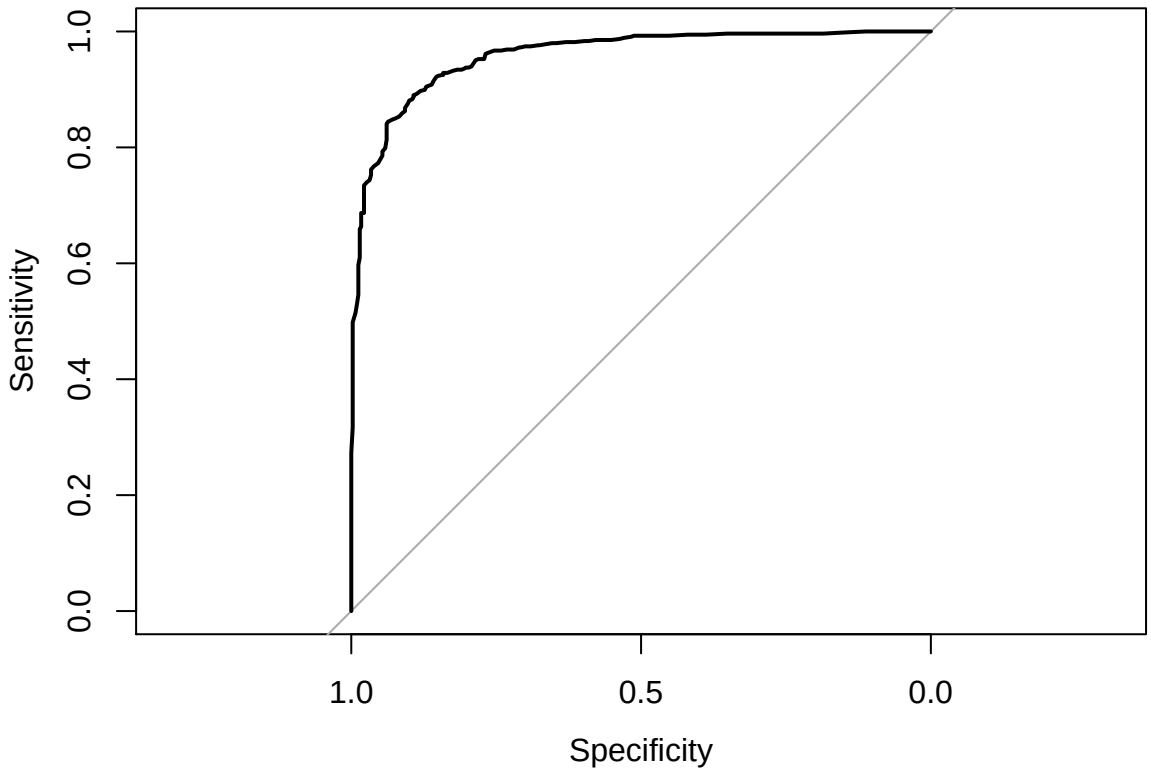
```
table(predTest,test1$cutMade)
```

```
##
## predTest   0   1
##        0 361  55
##        1  49 491
```

```
# predict on test and evaluate the model on test using auc-----------------------
# head(predTest)
# ?predict
pred_AUC <- predict(mod_forest, test1, type = "prob")[,1]

test1 <- test1 %>% mutate(prediction = pred_AUC)

roc_obj <- roc(test1$cutMade, test1$prediction)
auc(roc_obj)
```

```
## Area under the curve: 0.9583
```

```
plot(roc_obj)
```




```
#summary(roc_obj)
```

```
# Setting formula for random forest 2
f2 <- as.formula(cutMade ~ GIRsPerRound + ThreePuttsPerRound + over300DrivesPerRound + distPuttsMadePerR

# Training forest 2
mod_forest2 <- randomForest(f2, data = train1, ntree = 128, mtry = 2)
mod_forest2
```

```
##
## Call:
##  randomForest(formula = f2, data = train1, ntree = 128, mtry = 2)
##                Type of random forest: classification
##                      Number of trees: 128
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 16.65%
## Confusion matrix:
##      0    1 class.error
## 0 1342  293   0.1792049
## 1  344 1846   0.1570776
```

```
sum(diag(mod_forest2$confusion)) / nrow(train1)
```

```
## [1] 0.8334641
```

```
# Get importance, code from textbook
library(tibble)
importance(mod_forest2) %>%
  as.data.frame() %>%
  rownames_to_column() %>%
```

```
  arrange(desc(MeanDecreaseGini))
```

```
##                 rowname MeanDecreaseGini
## 1          GIRsPerRound         479.1078
## 2 distPuttsMadePerRound         420.7981
## 3    ThreePuttsPerRound         401.7033
## 4    proxToHoleApproach         300.2741
## 5 over300DrivesPerRound         251.6143
```

```
# https://www.r-bloggers.com/how-to-implement-random-forests-in-r/
predTest2 <- predict(mod_forest2, test1, type = "class")

mean(predTest2 == test1$cutMade)
```

```
## [1] 0.8441423
```

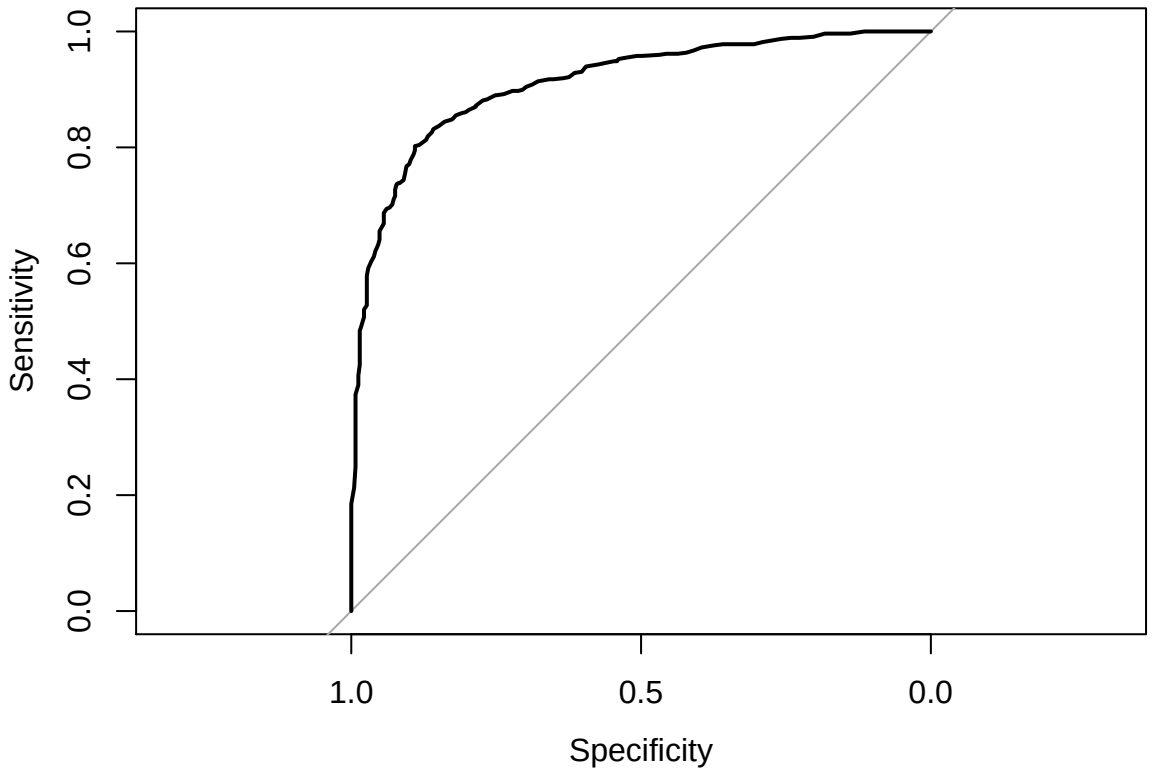
```
table(predTest2, test1$cutMade)
```

```
##
## predTest2   0   1
##         0 352  91
##         1  58 455
```

```
# predict on test and evaluate the model on test using auc-----------------------
# head(predTest2)
# ?predict
pred_AUC2 <- predict(mod_forest2, test1, type = "prob")[,1]

test1 <- test1 %>% mutate(prediction2 = pred_AUC2)

roc_obj <- roc(test1$cutMade, test1$prediction2)
auc(roc_obj)
```

```
## Area under the curve: 0.9113
```

```
plot(roc_obj)
```




```
#summary(roc_obj)
```

Below this is all old code.

```
# Neural Network 1

library(rpart)
library(partykit)

# Scale numeric data between 0 and 1
scale_0_1 <- function(x) {
  #' param x a numeric column that will be scaled
  rng <- range(x, na.rm = TRUE)
  (x - rng[1]) / (rng[2] - rng[1])
}

scaledData <- data.frame(lapply(dataFinal, FUN=function(x) if (is.numeric(x)) scale_0_1(x) else x))

###############
# split 80/20 -------------------------
set.seed(123)

n <- nrow(scaledData)
train_id <- sample(1:n, size=round(n*0.8)) # select approx 80% of the row numbers between 1 and n
train <- scaledData[train_id,] # the data set we'll train the model on
test <- scaledData[-train_id,] # the data set we'll test the model on

head(train)
```

```
##           Player.Name                  Event.Name cutMade birdiesPerRound
```

```
## 1375     Gooch, Talor                 Genesis Open       1       0.4642857
## 3769 Stallings, Scott           John Deere Classic       0       0.2142857
## 1955   Byrd, Jonathan                 Houston Open       1       0.3571429
## 4220   Kaymer, Martin              PGA Championship       1       0.3571429
## 4493 Grillo, Emiliano           THE NORTHERN TRUST       1       0.3928571
## 218    Malnati, Peter Sanderson Farms Championship       1       0.5000000
##      GIRsPerRound overParHolesPerRound ThreePuttsPerRound
## 1375    0.4600000            0.2888889          0.1250000
## 3769    0.8000000            0.1555556          0.0000000
## 1955    0.7066667            0.1555556          0.3333333
## 4220    0.7000000            0.1777778          0.1250000
## 4493    0.5800000            0.2000000          0.1875000
## 218     0.6400000            0.2444444          0.2500000
##      over300DrivesPerRound distPuttsMadePerRound proxToHoleApproach
## 1375            0.34615385             0.4971357          0.2489798
## 3769            0.30769231             0.4015811          0.3265320
## 1955            0.05128205             0.4289643          0.3063928
## 4220            0.03846154             0.5730981          0.2623529
## 4493            0.05769231             0.5976742          0.2770037
## 218             0.03846154             0.5641613          0.2338360
```

```
head(test)
```

```
##        Player.Name   Event.Name cutMade birdiesPerRound GIRsPerRound
## 1     Allan, Steve Safeway Open       0       0.1428571         0.68
## 4     Atkins, Matt Safeway Open       0       0.3571429         0.36
## 6  Baddeley, Aaron Safeway Open       0       0.4285714         0.56
## 11    Blixt, Jonas Safeway Open       1       0.4285714         0.50
## 21  Conners, Corey Safeway Open       1       0.3928571         0.70
## 24    Dahmen, Joel Safeway Open       0       0.5714286         0.64
##    overParHolesPerRound ThreePuttsPerRound over300DrivesPerRound
## 1             0.2444444             0.1250             0.2692308
## 4             0.3333333             0.0000             0.5769231
## 6             0.2444444             0.5000             0.1538462
## 11            0.2888889             0.0625             0.2307692
## 21            0.1555556             0.1875             0.4615385
## 24            0.4222222             0.3750             0.4230769
##    distPuttsMadePerRound proxToHoleApproach
## 1              0.3500229          0.5055532
## 4              0.6490605          0.2062307
## 6              0.4888863          0.2902794
## 11             0.4986824          0.3079143
## 21             0.4902612          0.2417792
## 24             0.6377177          0.3159097
```

```
# train a nueral network classifier using the same train data as above.
library(nnet)

f <- as.formula(cutMade ~ birdiesPerRound + GIRsPerRound + overParHolesPerRound + ThreePuttsPerRound)

train <- train %>%
  mutate(cutMade = as.integer(cutMade))

neuralNetwork <- nnet(f, data=train, size=3)
```

```
## # weights:  19
```

```
## initial  value 6952.330093
## final  value 2190.000000
## converged
```

```
# load library
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'
```

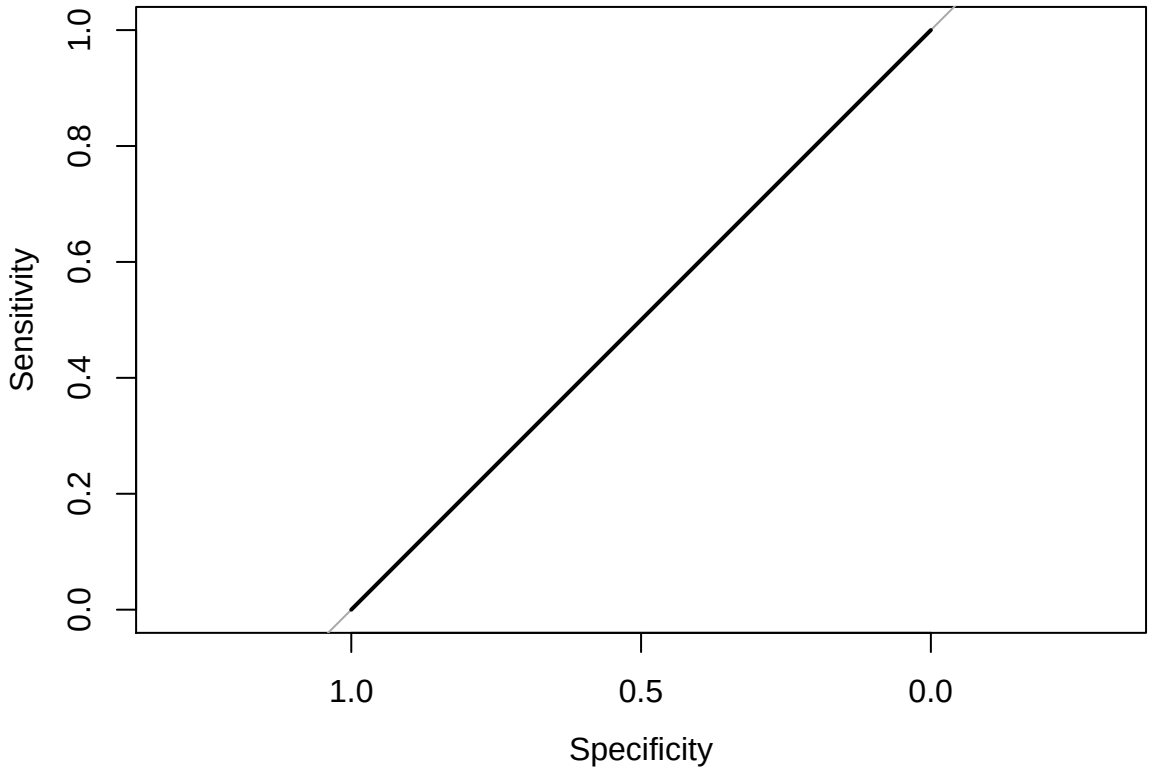
```
## The following object is masked from 'package:dplyr':
##
##     compute
```

```
# fit neural network
set.seed(2)
NN <- neuralnet(f, train, hidden = 1, linear.output = T)

# plot neural network
plot(NN)
```

```
# predict on test and evaluate the model on test using auc-----------------------
library(pROC)
pred <- predict(neuralNetwork, test, type = "raw")[,1] # a vector of probabilities
test <- test %>% mutate(prediction = pred)

roc_obj <- roc(test$cutMade, test$prediction)
auc(roc_obj)
```

```
## Area under the curve: 0.5
```

```
plot(roc_obj)
```



```
#summary(roc_obj)
```