Boxing Model

JR

August 15, 2018

In this project I am attempting to model whether a boxer would win, lose, or draw in a match purely based off of their physical metrics like age, height, reach, weight, and past results.

First, I'm calling the libraries I'll be using and reading in the csv file.

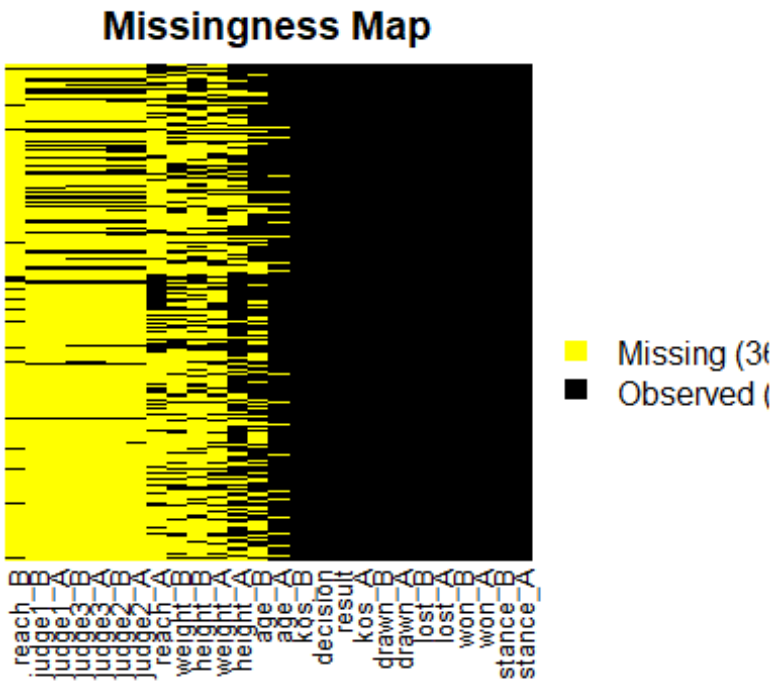
```
library(tidyr)
library(dplyr)
library(caTools)
library(nnet)
library(Amelia)

boxing <- read.csv('C:\\Users\\Jason\\Desktop\\R
Directory\\bouts_out_new.csv')
```

Next, I'll take a look at the data and check the missing values with a 'Missingness Map'

```
head(boxing)
```

```
##   age_A age_B height_A height_B reach_A reach_B stance_A stance_B weight_A
## 1    35    27      179      175     178     179 orthodox orthodox      160
## 2    26    31      175      185     179     185 orthodox orthodox      164
## 3    28    26      176      175      NA     179 orthodox orthodox      154
## 4    25    29      175      174     179     180 orthodox orthodox      155
## 5    25    35      175      170     179     170 orthodox orthodox      155
## 6    24    31      175      175     179     178 orthodox orthodox       NA
##   weight_B won_A won_B lost_A lost_B drawn_A drawn_B kos_A kos_B result
## 1      160    37    49      0      1       0       1    33    34   draw
## 2      164    48    50      1      2       1       1    34    32  win_A
## 3      154    23    47      0      1       1       1    13    33  win_B
## 4      155    46    31      1      3       1       0    32    19  win_A
## 5       NA    45    40      1      4       1       0    32    33  win_A
## 6       NA    44    32      1      1       1       0    31    28  win_A
##   decision judge1_A judge1_B judge2_A judge2_B judge3_A judge3_B
## 1       SD      110      118      115      113      114      114
## 2       UD      120      108      120      108      120      108
## 3       KO       NA       NA       NA       NA       NA       NA
## 4       KO       47       48       49       46       48       47
## 5       UD      118      110      119      109      117      111
## 6       KO       NA       NA       NA       NA       NA       NA
```

```
missmap(boxing, legend = TRUE, col = c('yellow', 'black'), y.labels = c(''),
y.at = c(1))
```



That's a lot of missing values. Since I am going to be using the boxer's physical attributes and past results to predict if they win or lose, I will delete the columns containing judge scores. Additionally, since each observation is a boxer it does not make sense to use dummy values for that boxer's physical attributes. Instead, I'll get rid of all of the missing values.

```
boxing <- boxing[,-(21:26)]
boxing <- na.omit(boxing)
str(boxing)
```

```
## 'data.frame':    7696 obs. of  20 variables:
##  $ age_A   : int  35 26 25 23 22 21 21 19 18 18 ...
##  $ age_B   : int  27 31 29 31 28 40 32 32 26 28 ...
##  $ height_A: int  179 175 175 175 175 175 175 175 175 175 ...
##  $ height_B: int  175 185 174 175 177 174 180 167 180 173 ...
##  $ reach_A : int  178 179 179 179 179 179 179 179 179 179 ...
##  $ reach_B : int  179 185 180 188 175 180 188 173 188 183 ...
##  $ stance_A: Factor w/ 3 levels "","orthodox",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ stance_B: Factor w/ 3 levels "","orthodox",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ weight_A: int  160 164 155 155 154 154 154 150 147 147 ...
##  $ weight_B: int  160 164 155 155 153 154 154 149 147 147 ...
##  $ won_A   : int  37 48 46 43 40 39 38 31 25 22 ...
##  $ won_B   : int  49 50 31 19 30 46 33 31 23 16 ...
##  $ lost_A  : int  0 1 1 1 0 0 0 0 0 0 ...
##  $ lost_B  : int  1 2 3 1 4 7 4 1 2 1 ...
##  $ drawn_A : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ drawn_B : int  1 1 0 2 0 1 1 1 0 0 ...
```

```
##  $ kos_A   : int  33 34 32 31 29 29 28 23 18 15 ...
##  $ kos_B   : int  34 32 19 12 18 39 28 23 21 15 ...
##  $ result  : Factor w/ 3 levels "draw","win_A",..: 1 2 2 2 2 2 2 2 2 2 ...
##  $ decision: Factor w/ 10 levels "DQ","KO","MD",..: 7 10 2 7 9 10 9 9 9 9
...
##  - attr(*, "na.action")= 'omit' Named int  3 5 6 8 9 10 14 15 16 17 ...
##   ..- attr(*, "names")= chr  "3" "5" "6" "8" ...
```

```
summary(boxing)
```

```
##      age_A           age_B            height_A        height_B
##  Min.   :15.00   Min.   :   0.00   Min.   :148.0   Min.   :150.0
##  1st Qu.:23.00   1st Qu.:  24.00   1st Qu.:169.0   1st Qu.:168.0
##  Median :26.00   Median :  28.00   Median :175.0   Median :175.0
##  Mean   :26.78   Mean   :  28.47   Mean   :175.9   Mean   :175.2
##  3rd Qu.:30.00   3rd Qu.:  32.00   3rd Qu.:183.0   3rd Qu.:181.0
##  Max.   :54.00   Max.   :2016.00   Max.   :213.0   Max.   :216.0
##
##     reach_A         reach_B          stance_A        stance_B
##  Min.   : 69.0   Min.   :  1.0           : 538           : 538
##  1st Qu.:173.0   1st Qu.:173.0   orthodox:5902   orthodox:5902
##  Median :180.0   Median :180.0   southpaw:1256   southpaw:1256
##  Mean   :181.4   Mean   :180.4
##  3rd Qu.:188.0   3rd Qu.:188.0
##  Max.   :427.0   Max.   :456.0
##
##     weight_A        weight_B         won_A            won_B
##  Min.   :104.0   Min.   :103.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:130.0   1st Qu.:130.0   1st Qu.: 14.00   1st Qu.: 14.00
##  Median :147.0   Median :146.5   Median : 24.00   Median : 23.00
##  Mean   :156.8   Mean   :157.3   Mean   : 30.97   Mean   : 28.08
##  3rd Qu.:174.0   3rd Qu.:175.0   3rd Qu.: 37.00   3rd Qu.: 35.00
##  Max.   :319.0   Max.   :334.0   Max.   :258.00   Max.   :223.00
##
##      lost_A           lost_B           drawn_A          drawn_B
##  Min.   : 0.000   Min.   :  0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:  2.000   1st Qu.: 0.000   1st Qu.: 0.000
##  Median : 2.000   Median :  4.000   Median : 0.000   Median : 1.000
##  Mean   : 3.962   Mean   :  7.236   Mean   : 1.781   Mean   : 2.112
##  3rd Qu.: 5.000   3rd Qu.: 10.000   3rd Qu.: 1.000   3rd Qu.: 2.000
##  Max.   :78.000   Max.   :102.000   Max.   :60.000   Max.   :58.000
##
##      kos_A            kos_B          result        decision
##  Min.   :  0.00   Min.   :  0.00   draw : 331   UD     :2397
##  1st Qu.:  8.00   1st Qu.:  7.00   win_A:6275   TKO    :2071
##  Median : 15.00   Median : 13.00   win_B:1090   KO     :1059
##  Mean   : 17.96   Mean   : 15.85                PTS    : 638
##  3rd Qu.: 24.00   3rd Qu.: 22.00                SD     : 505
##  Max.   :131.00   Max.   :121.00                MD     : 342
##                                                 (Other): 684
```

We still have more than 7,500 observations to work with. Next, I am going to clean up some of the data because the ranges don't make sense. For example, the minimum age of Boxer B cannot possibly be 0 years old. The following code cleans up those attributes. The ages are between 16 and 65, height is between 147cm and 214cm (the shortest boxer Jake Matlala and tallest boxer Nikolai Valuev), reach is between 160cm and 214m (longest reach is Sonny Liston), weight between 95 and 323 (no boxer being below 95 lbs and the heaviest boxer being Nikolai Valuev)

```
boxing <- subset(boxing, age_A >= 16 & age_A <= 65)
boxing <- subset(boxing, age_B >= 16 & age_B <= 65)
boxing <- subset(boxing, height_A >= 147 & height_A <= 214)
boxing <- subset(boxing, height_B >= 147 & height_B <= 214)
boxing <- subset(boxing, reach_A >= 160 & reach_A <= 214)
boxing <- subset(boxing, reach_B >= 160 & reach_B <= 214)
boxing <- subset(boxing, weight_A >= 95 & weight_A <= 323)
boxing <- subset(boxing, weight_B >= 95 & weight_B <= 323)
```

Now that our data is within normal parameters I'll start modelling. First, I'll divide the data into train/test splits

```
sample <- sample.split(boxing$result, SplitRatio = 0.8)
boxing_train <- subset(boxing, sample == TRUE)
boxing_test <- subset(boxing, sample == FALSE)
```

I'm going to use a multinomial model from the nnet package because I am trying to determine one out of three possible scenarios: win, lose, or draw. I am subtracting the 'decision' column from being used in the model because I want this to be unsupervised.

```
model <- multinom(result ~ .-decision, data = boxing_train)

## # weights:  66 (42 variable)
## initial  value 6479.615279
## iter  10 value 3269.643212
## iter  20 value 3251.084611
## iter  30 value 3242.024977
## iter  40 value 3094.413684
## final  value 3093.802217
## converged
```

Now that the model is up I'll use it to predict from our test split.

```
predict_result <- predict(model, boxing_test)
```

I'll use a confusion matrix and misclassification error calculation to see how the model did compared to the actual results.

```
table(predict_result, boxing_test$result)

##
## predict_result draw win_A win_B
##           draw     0     1     1
```

```
##          win_A   60  1191   198
##          win_B    3    10    11
```

```
mean(as.character(predict_result)!= as.character(boxing_test$result))
```

```
## [1] 0.1850847
```

This model seems to be very good at predicting if Boxer A would win but not very good at predicting boxer B's wins.