# George_Smith_IST707_Project

George Smith

8/9/2021

## Introduction

Sumo wrestling is a competitive form of wrestling, where rikishi (fighter) look to push their opponent out of the dohyo (circular ring) or onto the ground. Sumo originated in Japan and is the only country where it practiced professionally. As a result a majority of the world is not familiar with sumo, and would not know the first thing about selecting a potential winner of a sumo match. The purpose of this project is to use data mining techniques to predict the potential winner of a sumo match.

## About the data

The data is an aggregation results of every bout (top two divisions) in Japanese sumo wrestling grand tournaments (honbasho) from 1985 until 2019 scraped from http://sumodb.sumogames.de/Results.aspx?b=198301&d=15&simple=on. The data can be found here in spreadsheet format https://data.world/cervus/sumo-results

#install packages

```
if(!require("tm")) {install.packages("tm")}
```

```
## Loading required package: tm
```

```
## Loading required package: NLP
```

```
if(!require("stringr")) {install.packages("stringr")}
```

```
## Loading required package: stringr
```

```
if(!require("stringi")) {install.packages("sttringi")}
```

```
## Loading required package: stringi
```

```
if(!require("Matrix")) {install.packages("Matrix")}
```

```
## Loading required package: Matrix
```

```r
if(!require("rpart")) {install.packages("rpart")}
```

```
## Loading required package: rpart
```

```r
if(!require("rpart.plot")) {install.packages("rpart.plot")}
```

```
## Loading required package: rpart.plot
```

```r
if(!require("rattle")) {install.packages("rattle")}
```

```
## Loading required package: rattle
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
##
## Attaching package: 'bitops'
```

```
## The following object is masked from 'package:Matrix':
##
##     %&%
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
if(!require("RColorBrewer")) {install.packages("RColorBrewer")}
```

```
## Loading required package: RColorBrewer
```

```r
if(!require("ggplot2")) {install.packages("ggplot2")}
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```r
if(!require("neuralnet")) {install.packages("neuralnet")}
```

```
## Loading required package: neuralnet
```

```
if(!require("fastDummies")) {install.packages("fastDummies")}
```

```
## Loading required package: fastDummies
```

## libraries

```
library(stringr)
library(stringi)
library(Matrix)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:neuralnet':
##
##     compute

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(rpart)
library(rattle)
library(rpart.plot)
library(RColorBrewer)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
library(dplyr)
library(neuralnet)
library(fastDummies)
```

## read in data

```
sumo <- read.csv("C:/Users/GeorgeSmith/Documents/IST 707 Project/2019.csv")

str(sumo)
```

```
## 'data.frame':    4990 obs. of  13 variables:
##  $ basho           : num  2019 2019 2019 2019 2019 ...
##  $ day             : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ rikishi1_id     : int  6572 12231 11971 12255 4980 7239 1284 12113 12024 12203 ...
##  $ rikishi1_rank   : chr  "J14e" "Ms1w" "J13w" "J14w" ...
##  $ rikishi1_shikona: chr  "Gagamaru" "Kiribayama" "Jokoryu" "Chiyonoumi" ...
##  $ rikishi1_result : chr  "1-0 (8-7)" "0-1 (4-3)" "0-1 (5-9-1)" "1-0 (8-5-2)" ...
##  $ rikishi1_win    : int  1 0 0 1 0 1 0 1 1 0 ...
##  $ kimarite        : chr  "oshitaoshi" "oshitaoshi" "yorikiri" "yorikiri" ...
##  $ rikishi2_id     : int  12231 6572 12255 11971 7239 4980 12113 1284 12203 12024 ...
##  $ rikishi2_rank   : chr  "Ms1w" "J14e" "J14w" "J13w" ...
##  $ rikishi2_shikona: chr  "Kiribayama" "Gagamaru" "Chiyonoumi" "Jokoryu" ...
##  $ rikishi2_result : chr  "0-1 (4-3)" "1-0 (8-7)" "1-0 (8-5-2)" "0-1 (5-9-1)" ...
##  $ rikishi2_win    : int  0 1 1 0 1 0 1 0 0 1 ...
```

## view the Data Frame

```
head(sumo)
```

```
##     basho day rikishi1_id rikishi1_rank rikishi1_shikona rikishi1_result
## 1 2019.01   1        6572          J14e         Gagamaru       1-0 (8-7)
## 2 2019.01   1       12231          Ms1w       Kiribayama       0-1 (4-3)
## 3 2019.01   1       11971          J13w          Jokoryu     0-1 (5-9-1)
## 4 2019.01   1       12255          J14w       Chiyonoumi     1-0 (8-5-2)
## 5 2019.01   1        4980          J12w        Sokokurai       0-1 (8-7)
## 6 2019.01   1        7239          J13e       Kyokushuho       1-0 (9-6)
##   rikishi1_win   kimarite rikishi2_id rikishi2_rank rikishi2_shikona
## 1            1 oshitaoshi       12231          Ms1w       Kiribayama
## 2            0 oshitaoshi        6572          J14e         Gagamaru
## 3            0   yorikiri       12255          J14w       Chiyonoumi
## 4            1   yorikiri       11971          J13w          Jokoryu
## 5            0   yorikiri        7239          J13e       Kyokushuho
## 6            1   yorikiri        4980          J12w        Sokokurai
##   rikishi2_result rikishi2_win
## 1       0-1 (4-3)            0
## 2       1-0 (8-7)            1
## 3     1-0 (8-5-2)            1
## 4     0-1 (5-9-1)            0
## 5       1-0 (9-6)            1
## 6       0-1 (8-7)            0
```

## Analyze the columns

```
unique(sumo$basho)
```

```
## [1] 2019.01 2019.03 2019.05 2019.07 2019.09
```

```
unique(sumo$day)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
```

## Used the dplyr package to drop the rikishi2_win, rikishi1_result,and rikishi2_result column because their relationship to rikishi1_win is obvious and leads to a poor model

```
sumo <-select (sumo,-c(rikishi2_win,rikishi1_result,rikishi2_result, ))
```

## Need to exclude the below records as they do not have enought values to be run in the models

```
sumo <- sumo[!(sumo$rikishi1_rank=="Ms5e"|
sumo$rikishi1_rank=="Ms5w" |
sumo$rikishi1_rank=="Ms2w" |
sumo$rikishi1_rank=="Ms3w" |
sumo$rikishi1_rank=="Ms4e" |
sumo$rikishi1_rank=="Ms4w" |
sumo$rikishi2_rank=="Ms5w" |
sumo$rikishi2_rank=="Ms2w" |
sumo$rikishi2_rank=="Ms3w" |
sumo$rikishi2_rank=="Ms4e" |
sumo$rikishi2_rank=="Ms5e" |
sumo$rikishi2_rank=="Ms4w") ,]
```

## Need to remove these values as well to run models

```
sumo <- sumo[!(sumo$rikishi1_shikona =="Chiyonoo"| sumo$rikishi1_shikona=="Kotodaigo" |
sumo$rikishi1_shikona =="Chiyonoo" |
sumo$rikishi1_shikona =="Kotokuzan"|
sumo$rikishi1_shikona=="Kototebakari"|
sumo$rikishi1_shikona == "Akua"|
sumo$rikishi1_shikona == "Nishikifuji"|
sumo$rikishi1_shikona == "Tamaki" |
sumo$kimarite == "okurigake" |
sumo$rikishi2_shikona == "Akua" |
sumo$rikishi2_shikona == "Tamaki" ) ,]
```

# view the first records of out data frame

```
head(sumo)
```

```
##     basho day rikishi1_id rikishi1_rank rikishi1_shikona rikishi1_win
## 1 2019.01   1        6572          J14e         Gagamaru            1
## 2 2019.01   1       12231          Ms1w       Kiribayama            0
## 3 2019.01   1       11971          J13w          Jokoryu            0
## 4 2019.01   1       12255          J14w       Chiyonoumi            1
## 5 2019.01   1        4980          J12w        Sokokurai            0
## 6 2019.01   1        7239          J13e       Kyokushuho            1
##     kimarite rikishi2_id rikishi2_rank rikishi2_shikona
## 1 oshitaoshi       12231          Ms1w       Kiribayama
## 2 oshitaoshi        6572          J14e         Gagamaru
## 3   yorikiri       12255          J14w       Chiyonoumi
## 4   yorikiri       11971          J13w          Jokoryu
## 5   yorikiri        7239          J13e       Kyokushuho
## 6   yorikiri        4980          J12w        Sokokurai
```

### run the decision tree

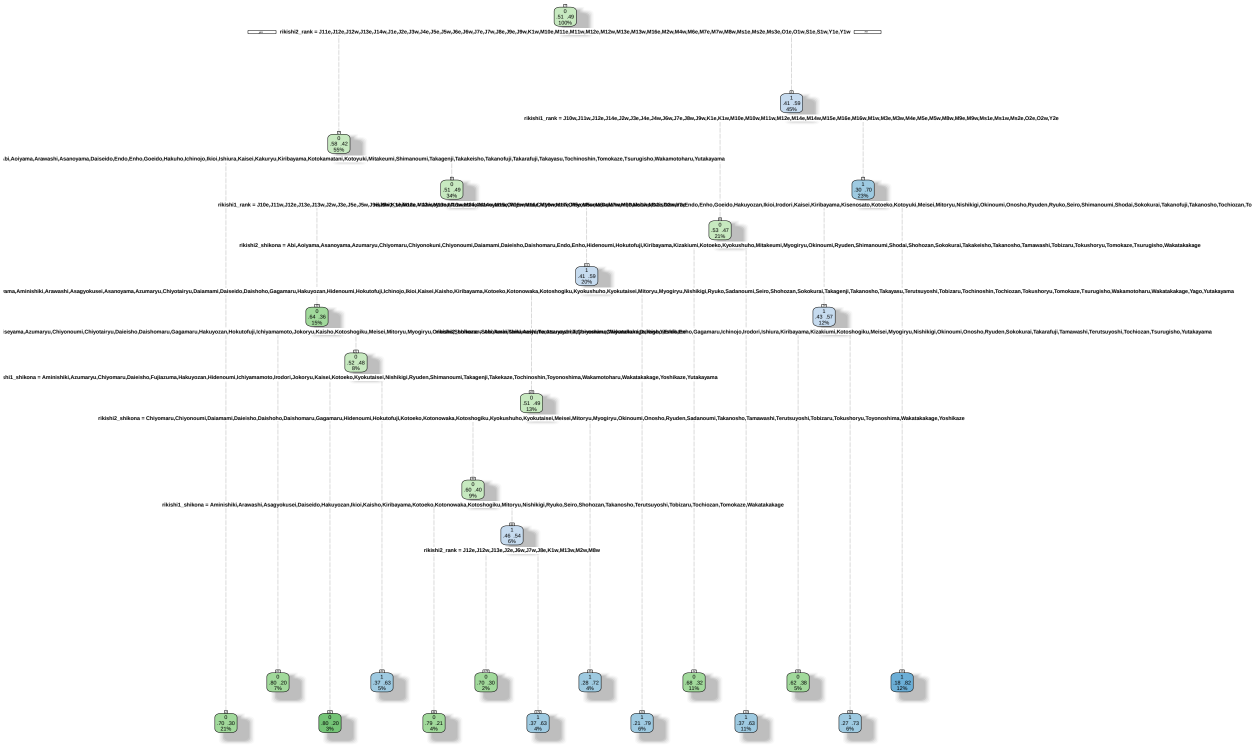
# set the training ratio

```
trainRatio <- .60

set.seed(11) # Set Seed so that same sample can be reproduced in future also

# create the training and testing data
sample <- sample(1:nrow(sumo), trainRatio*nrow(sumo), replace = F)
dfTrain <- sumo[sample,]
dfTest <- sumo[-sample,]

# run the decision tree
tree.train <- rpart(rikishi1_win ~ ., data = dfTrain, method="class",
                    minsplit = 2, minbucket = 1)
```

#plot the decision tree

```
fancyRpartPlot(tree.train)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```


0
.51 .49
100%
yes
rikishi2_rank = J11e,J12e,J12w,J13e,J14w,J1e,J2e,J3w,J4e,J5e,J5w,J6e,J6w,J7e,J7w,J8e,J9e,J9w,K1w,M10e,M11e,M11w,M12e,M12w,M13e,M13w,M16e,M2w,M4w,M6e,M7e,M7w,M8w,Ms1e,Ms2e,Ms3e,O1e,O1w,S1e,S1w,Y1e,Y1w
no
1
.41 .59
45%
rikishi1_rank = J10w,J11w,J12e,J14e,J2w,J3e,J4e,J4w,J6w,J7e,J8w,J9w,K1e,K1w,M10e,M10w,M11w,M12e,M14e,M14w,M15e,M16e,M16w,M1w,M3e,M3w,M4e,M5e,M5w,M8w,M9e,M9w,Ms1e,Ms1w,Ms2e,O2e,O2w,Y2e
0
.58 .42
55%
bi,Aoiyama,Arawashi,Asanoyama,Daiseido,Endo,Enho,Goeido,Hakuho,Ichinojo,Ikioi,Ishiura,Kaisei,Kakuryu,Kiribayama,Kotokamatani,Kotoyuki,Mitakeumi,Shimanoumi,Takagenji,Takakeisho,Takanofuji,Takarafuji,Takayasu,Tochinoshin,Tomokaze,Tsurugisho,Wakamotoharu,Yutakayama
0
.51 .49
34%
1
.30 .70
23%
Endo,Enho,Goeido,Hakuyozan,Ikioi,Irodori,Kaisei,Kiribayama,Kisenosato,Kotoeko,Kotoyuki,Meisei,Mitoryu,Nishikigi,Okinoumi,Onosho,Ryuden,Ryuko,Seiro,Shimanoumi,Shodai,Sokokurai,Takanofuji,Takanosho,Tochiozan,To
0
.53 .47
21%
rikishi2_shikona = Abi,Aoiyama,Asanoyama,Azumaryu,Chiyomaru,Chiyonokuni,Chiyonoumi,Daiamami,Daieisho,Daishomaru,Endo,Enho,Hidenoumi,Hokutofuji,Kiribayama,Kizakiumi,Kotoeko,Kyokushuho,Mitakeumi,Myogiryu,Okinoumi,Ryuden,Shimanoumi,Shodai,Shohozan,Sokokurai,Takakeisho,Takanosho,Tamawashi,Tobizaru,Tokushoryu,Tomokaze,Tsurugisho,Wakatakakage
1
.41 .59
20%
ama,Aminishiki,Arawashi,Asagyokusei,Asanoyama,Azumaryu,Chiyotairyu,Daiamami,Daiseido,Daishoho,Gagamaru,Hakuyozan,Hidenoumi,Hokutofuji,Ichinojo,Ikioi,Kaisei,Kaisho,Kiribayama,Kotoeko,Kotonowaka,Kotoshogiku,Kyokushuho,Kyokutaisei,Mitoryu,Myogiryu,Nishikigi,Ryuko,Sadanoumi,Seiro,Shohozan,Sokokurai,Takagenji,Takanosho,Takayasu,Terutsuyoshi,Tobizaru,Tochinoshin,Tochiozan,Tokushoryu,Tomokaze,Tsurugisho,Wakamotoharu,Wakatakakage,Yago,Yutakayama
0
.64 .36
15%
1
.43 .57
12%
Gagamaru,Ichinojo,Irodori,Ishiura,Kiribayama,Kizakiumi,Kotoshogiku,Meisei,Myogiryu,Nishikigi,Okinoumi,Onosho,Ryuden,Sokokurai,Takarafuji,Tamawashi,Terutsuyoshi,Tochiozan,Tsurugisho,Yutakayama
0
.52 .48
8%
hi1_shikona = Aminishiki,Azumaryu,Chiyomaru,Daieisho,Fujiazuma,Hakuyozan,Hidenoumi,Ichiyamamoto,Irodori,Jokoryu,Kaisei,Kotoeko,Kyokutaisei,Nishikigi,Ryuden,Shimanoumi,Takagenji,Takekaze,Tochinoshin,Toyonoshima,Wakamotoharu,Wakatakakage,Yoshikaze,Yutakayama
0
.51 .49
13%
rikishi2_shikona = Chiyomaru,Chiyonoumi,Daiamami,Daieisho,Daishoho,Daishomaru,Gagamaru,Hidenoumi,Hokutofuji,Kotoeko,Kotonowaka,Kotoshogiku,Kyokushuho,Kyokutaisei,Meisei,Mitoryu,Myogiryu,Okinoumi,Onosho,Ryuden,Sadanoumi,Takanosho,Tamawashi,Terutsuyoshi,Tobizaru,Tokushoryu,Toyonoshima,Wakatakakage,Yoshikaze
0
.60 .40
9%
rikishi1_shikona = Aminishiki,Arawashi,Asagyokusei,Daiseido,Hakuyozan,Ikioi,Kaisho,Kiribayama,Kotoeko,Kotonowaka,Kotoshogiku,Mitoryu,Nishikigi,Ryuko,Seiro,Shohozan,Takanosho,Terutsuyoshi,Tobizaru,Tochiozan,Tomokaze,Wakatakakage
1
.46 .54
6%
rikishi2_rank = J12e,J12w,J13e,J2e,J6w,J7w,J8e,K1w,M13w,M2w,M8w
0
.80 .20
7%
1
.37 .63
5%
0
.70 .30
2%
1
.28 .72
4%
0
.68 .32
11%
0
.62 .38
5%
1
.18 .82
12%
0
.70 .30
21%
0
.80 .20
3%
0
.79 .21
4%
1
.37 .63
4%
1
.21 .79
6%
1
.37 .63
11%
1
.27 .73
6%

Rattle 2021−Sep−19 17:42:08 GeorgeSmith

# predict against the test data

```
tree.pred <- predict(tree.train, dfTest, type="class")

summary(tree.pred)
```

```
##   0   1 
## 987 993
```

## predict if a fighter will win a given matchup using decision tree

```
with(dfTest, table(tree.pred,rikishi1_win ))
```

```
##          rikishi1_win
## tree.pred   0   1
##         0 528 459
##         1 443 550
```

Using the decision tree approach we were able to predict 528 instances of losses and 550 instances of wins correctly. This means that the decision tree model was able to predict sumo wrestling losses with 54% accuracy and wins with 56 % accuracy.

## Run neural network

```
head(sumo)
```

```
##     basho day rikishi1_id rikishi1_rank rikishi1_shikona rikishi1_win
## 1 2019.01   1        6572          J14e         Gagamaru            1
## 2 2019.01   1       12231          Ms1w       Kiribayama            0
## 3 2019.01   1       11971          J13w          Jokoryu            0
## 4 2019.01   1       12255          J14w       Chiyonoumi            1
## 5 2019.01   1        4980          J12w        Sokokurai            0
## 6 2019.01   1        7239          J13e       Kyokushuho            1
##     kimarite rikishi2_id rikishi2_rank rikishi2_shikona
## 1 oshitaoshi       12231          Ms1w       Kiribayama
## 2 oshitaoshi        6572          J14e         Gagamaru
## 3   yorikiri       12255          J14w       Chiyonoumi
## 4   yorikiri       11971          J13w          Jokoryu
## 5   yorikiri        7239          J13e       Kyokushuho
## 6   yorikiri        4980          J12w        Sokokurai
```

## Used the fast dummies package to make categorical variables numerical

```
transformed_data_nn <- dummy_cols(sumo, select_columns = c("rikishi1_rank","rikishi1_shikona", "kimarit
```

## drop original column as dummy variables exist

```
transformed_data_nn <-select (transformed_data_nn,-c(rikishi1_rank,rikishi1_shikona,kimarite, rikishi2_
```

## Making sure the above code worked

```
#head(transformed_data_nn)
```

## create training and testing data

```
trainRatio <- .60

set.seed(11) # Set Seed so that same sample can be reproduced in future also

# create the training and testing data
sample <- sample(1:nrow(transformed_data_nn), trainRatio*nrow(transformed_data_nn), replace = F)
dfTrainNN <- transformed_data_nn[sample,]
dfTestNN <- transformed_data_nn[-sample,]
```

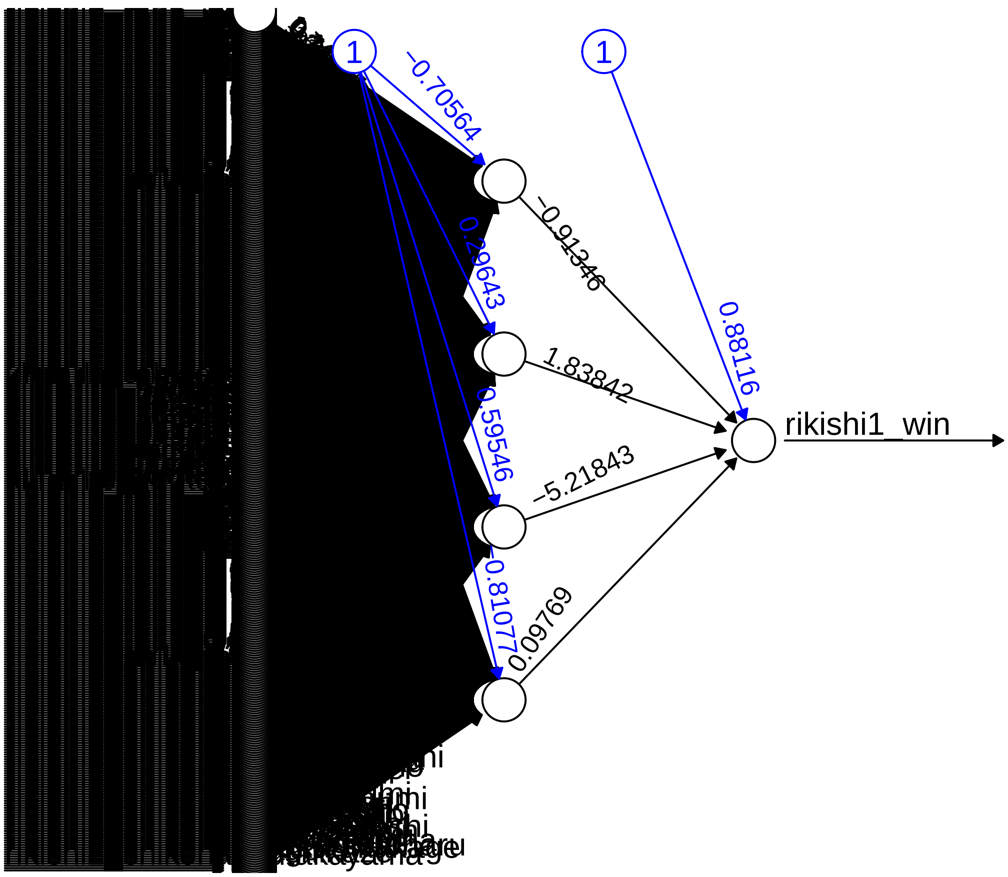
#run the NN

```
sumonet <- neuralnet(rikishi1_win ~ ., data = dfTrainNN , hidden=4, lifesign="minimal", linear.output =
```

```
## hidden: 4    thresh: 0.01    rep: 1/1    steps:      92  error: 370.74733    time: 1.78 secs
```

## NN plot

```
plot(sumonet, rep="best")
```



## predict against the test data

```
NN.pred <- predict(sumonet, dfTestNN, type="class")

table(NN.pred)
```

```
## NN.pred
##   0.4919260577686 0.491926057768601 0.491926057768603  0.49192605776861
##              1785                 1                 1                 1
## 0.491926058084983 0.491926058321837 0.491926384658911 0.491927548185853
##                 1                 1                 1                 1
## 0.516284283445566 0.516301109487379 0.516323657174952 0.516338724755133
##                 1                 1                 1                 1
## 0.516340769990688 0.516341137039421
##                 1               183
```

#round the predictions to be either win (1) or loss (0)

```
NNPredRound<-round(NN.pred)
```

## make predictions with the NN

```
with(dfTestNN, table(NNPredRound,rikishi1_win ))
```

```
##            rikishi1_win
## NNPredRound   0   1
##           0 876 916
##           1  95  93
```

Using the Neural Network approach we were able to correctly predict 876 losses correctly and 93 wins correctly. 90% of the loss predictions were correct, compared to only 9% of the win predictions being correct. I believe these poor results are due to having the create dummy variables for a majority of the data set so that the data would be usable for the Neural Network Model. I believe models such as the decision tree that are able to use categorical data would be better suited to make predictions on the Sumo data set.

# Conclusion

Using machine learning techniques including Decision Tree and Neural Network Models we were able to successfully predict if a sumo wrestler would win or lose a given match. Based on the above analysis it appears that the Decision Tree model was better suited for the Sumo data used as the Neural Network was able to only predict 9% of wins correctly. Although 56% win prediction accuracy is not incredible it is an improvement over simply guessing the result of a given match.