Win Prediction

OBJECTIVE:

Here I will use a neural net and SVM try to determine whether a basketball team's stats can help predict how much they will win or lose by in a game.

Here we will load the libraries we will be using for the experiments

```
library(neuralnet)
library(e1071)
library(kernlab)
```

He we will segment the data for use later

```
# GET SPECIFIC DATA

file <- 0 # the file we will be loading
all_data <- 0 # subset of file with columns that will actually be used,
              # and are complete cases + win_difference
home_stats <- 0 # subset of all_data containing only home team stats (except points)
away_stats <- 0 # subset of all_data containing only away team stats (except points)
win_difference <- 0 # home teams score - away team score
home_team_wins <- 0 # basically a binary of win_difference (factored)

file <- read.csv("games.csv", header=TRUE, stringsAsFactors = FALSE) #load data
all_data <- file[complete.cases(c("FG_PCT_home","FT_PCT_home","FG3_PCT_home","AST_home",
                                  "REB_home","HOME_TEAM_WINS","PTS_home","PTS_away",
                                  "FG_PCT_away","FT_PCT_away","FG3_PCT_away",
                                  "AST_away",   "REB_away")), ]
all_data$win_difference=(all_data$PTS_home-all_data$PTS_away)
home_stats <- all_data[c("FG_PCT_home","FT_PCT_home","FG3_PCT_home","AST_home","REB_home")]
away_stats <- all_data[c("FG_PCT_away","FT_PCT_away","FG3_PCT_away","AST_away","REB_away")]
win_difference <- all_data$win_difference
home_team_wins <- all_data$HOME_TEAM_WINS
all_data$HOME_TEAM_WINS<-factor(all_data$HOME_TEAM_WINS,levels=c(0,1),labels=c("W","L"))

n_data <- all_data[1:2000, c("FG_PCT_home","FT_PCT_home","FG3_PCT_home","AST_home",
                                  "REB_home",
                                  "FG_PCT_away","FT_PCT_away","FG3_PCT_away",
                                  "AST_away",   "REB_away","win_difference")]


ref_data <- all_data[1:20000, c("FG_PCT_home","FT_PCT_home","FG3_PCT_home","AST_home",
                                  "REB_home",
                                  "FG_PCT_away","FT_PCT_away","FG3_PCT_away",
                                  "AST_away",   "REB_away","win_difference")]
```

Now we will normalize the data set we will be using, and train the neural net on a segment of it. We will be using 1000 observations and 5 nodes on 1 layer for the neural network for this test

```
normalize <- function(x) {
    return((x - min(x)) / (max(x) - min(x)))
}

n_data <- as.data.frame(lapply(n_data, normalize))

win_model<-neuralnet(win_difference ~ . , data = n_data[1:1000, ], hidden=5)

plot(win_model)

model_results <- compute (win_model,n_data[1001:2000, ])
predicted_difference <- model_results$net.result

cor (predicted_difference, n_data$win_difference[1001:2000])
```

```
##           [,1]
## [1,] 0.8683057
```

Now we will continue to try and improve the model. We will attempt to do this by using a softplus activator function, we will also use 5 nodes on 2 layers for this test

```
softplus <- function(x) { log(1 + exp(x)) } #new activator function

win_model_2 <-neuralnet(win_difference ~ . , data = n_data[1:1000, ], hidden=c(5,5), act.fct = softplus)

model_results_2 <- compute (win_model_2,n_data[1001:2000, ])
predicted_difference_2 <- model_results$net.result

cor (predicted_difference_2, n_data$win_difference[1001:2000])
```

```
##           [,1]
## [1,] 0.8683057
```

```
plot(win_model_2)
```

The changes we made to the model didn't seem to meaningfully change success rate of the algorithm. 87% is a decent success rate for our purposes. We will now try to use an SVM to see how it will compare to our neural network success rate.

```
u_data <- ref_data[c("FG_PCT_home","FG_PCT_away","win_difference")]

u_data$win_difference <- factor(u_data$win_difference)

svm_model <- ksvm(win_difference ~ . , data = u_data[1:1000, ])
svm_predict <- predict (svm_model, u_data[1001:2000, ])

agreement <- svm_predict == u_data[1001:2000, ]$win_difference
table(agreement)
```

```
## agreement
## FALSE  TRUE
##   992     8
```

This result is horrible for the Support Vector Model. It does not seem to have any success predicting the absolute win difference based on the given observations.

so we will run the same experiment, but instead of seeing whether it can come close to detecting the absolute

difference in score between the two teams, we will see if it can test it within a range. For example, whether the teams will be within 10 points of each other. we will also add more variables, and use a larger data set.

```
u_data <- ref_data#[c("FG_PCT_home","FG_PCT_away","win_difference")]

u_data$win_difference <- abs(u_data$win_difference) < 10
u_data$win_difference <- factor(u_data$win_difference)

examples_num <- 10000
train_num <- 7500

svm_model <- ksvm(win_difference ~ . , data = u_data[1:train_num, ])
svm_predict <- predict (svm_model, u_data[(train_num+1) : examples_num, ])


table(svm_predict,u_data[(train_num+1):examples_num,"win_difference"])
```

```
##
## svm_predict FALSE TRUE
##       FALSE   744  205
##       TRUE    481 1070
```

```
agreement <- svm_predict == u_data[(train_num+1):examples_num, ]$win_difference
table(agreement)
```

```
## agreement
## FALSE  TRUE
##   686  1814
```

```
prop.table(table(agreement))
```

```
## agreement
##  FALSE   TRUE
## 0.2744 0.7256
```

Making these changes, the model is now correct about 73% of the time.

We will now try changing the cost values to see if we can get a better success rate

```
x=100

set.seed(12345)

#train support vector algorithm
m <- ksvm(win_difference ~ . , data = u_data[1:train_num, ], kernal="rbfdot",C=x)

#test against prediction for other segment of data
pred <- predict (m, u_data[(train_num+1) : examples_num, ])

table(pred,u_data[(train_num+1):examples_num,"win_difference"])
```

```
##
## pred    FALSE TRUE
##   FALSE   751  303
##   TRUE    474  972
```

```
agreement <- pred == u_data[(train_num+1):examples_num, ]$win_difference
table(agreement)
```

```
## agreement
## FALSE  TRUE
##   777  1723
```

```
prop.table(table(agreement))
```

```
## agreement
##  FALSE   TRUE
## 0.3108 0.6892
```

After trying different cost values, the success rate of the model goes down as it gets larger.

In conclusion, our neural network had success predicting the point difference between teams by using their stats, but it looks like the SVM was only able to predict the win difference within a wide range.