SEAHAWKS SHOULD HAVE RAN THE BALL

Artificial American Football Coach with Decision Trees

Benjamin Jones

February 28, 2020

Introduction

Have you ever dreamed of being an NFL coach? No? Me neither... Maybe you have a child on a pee wee football team in need of a coach? Also no? Same... But perhaps you like data? If *you're* reading this, chances are you answered 'yes' to at least one of those questions.

The following dataset consists of football situations and the correct play to call for each situation. Let's see if it is possible to create a decision tree which, given a situation, can output the correct play to call. We'll try a few different tree models. The dataset is titled *Football Strategy* and publicly available here.

Setup

Load the necessary libraries as well as authenticate the data.world API:

```
library(data.world)
library(jsonlite)
library(stringr)
library(rpart)
library(rpart.plot)
library(ipred)
library(randomForest)
library(rpart)
library(caret)
setwd("~/Desktop")
dw_auth <- read_json("dw_cred.json")
saved_cfg <- save_config(dw_auth$token)
set_config(saved_cfg)
```

Query the data:

```
df <- query(qry_sql(paste0("SELECT football_scenarios_dfe_832307.antecedent as LABEL,
                            football_scenarios_dfe_832307.orig_antecedent as SITUATION ",
                            "FROM football_scenarios_dfe_832307")),
                            dataset="https://data.world/wppdatacatalogue/football-strategy")
```

Preview the data:

```
head(df, 3)
```

```
## # A tibble: 3 x 2
##   LABEL           SITUATION
##   <chr>           <chr>
## 1 kick a field g~ It is first down and 10. The ball is on your opponent's 20 ya~
## 2 kick a field g~ It is second down and inches. The ball is on your opponent's ~
## 3 kick a field g~ It is second down and inches. The ball is on your opponent's ~
```

It's difficult to see, but an example of a full value of an entry in **SITUATION** is "It is first down and 10. The ball is on your opponent's 20 yardline. There is 3 seconds left in the second quarter. You are down by 3 points. Would you:"

Cleaning

The entries of the football situation is a messy, as it is a long string. Fortunately, the values are consistent and we can do use patterns to pull out useful quantitative and qualitative features. For example:

"It is first down and 10. The ball is on your opponent's 20 yardline. There is 3 seconds left in the second quarter. You are down by 3 points. Would you:"

"It is third down and inches. The ball is on your 15 yardline. There is 3 seconds left in the fourth quarter. You are up by 7 points. Would you:"

Numerical values used will be: yards to go, yard line, seconds left, and point differential. When distance is 'inches', it will be converted to 0.

Categorical valeus used will be: down and quarter. ### Remove Cases that Don't Follow the Pattern

```
df <- df[grep("It is", df$SITUATION),]
df <- df[grep("The ball is", df$SITUATION),]
df <- df[grep("There is", df$SITUATION),]
```

Down and Yards to Go

```
df$DOWN <- ""
df$YTG <- ""

for (i in 1:nrow(df)) {
ss1 <- str_split_fixed(df[i,2], " down ", 2)[1,1]
df$DOWN[i] <- str_split_fixed(ss1, "It is ", 2)[1,2]

ss2 <- str_split_fixed(df[i,2], " down ", 2)[1,2]
ss3 <- str_split_fixed(ss2, "and ", 2)
df$YTG[i] <- str_split_fixed(ss3, "\\.", 2)[2,1]
}

df$YTG <- str_replace(df$YTG, "inches", "0")
df$YTG <- as.numeric(df$YTG)
```

Yardline

```
df$YL <- 0
for (i in 1:nrow(df)) {
ss1 <- str_split_fixed(df[i,2], "The ball is on ", 2)[1,2]
```

```
ss2 <- str_split_fixed(ss1, " yardline.", 2)[1,1]
df$YL[i] <- as.numeric(gsub("[^0-9.-]", "", ss2))
  if (grepl("your opponent", ss1)) {
    df$YL[i] <- 50 - df$YL[i]
  } else {
    df$YL[i] <- df$YL[i] - 50
  }
}
```

Seconds and Quarter

```
df$SEC <- 0
df$Q <- ""
for (i in 1:nrow(df)) {
  ss1 <- str_split_fixed(df[i,2], "There is ", 2)[1,2]
  if (grepl("minute", ss1)) {
    df$SEC[i] <- as.numeric(str_split_fixed(ss1, " minute", 2)[1,1])*60
  } else{
    df$SEC[i] <- as.numeric(str_split_fixed(ss1, " second", 2)[1,1])
  }
  ss2 <- str_split_fixed(ss1, " quarter", 2)[1,1]
  df$Q[i] <- str_split_fixed(ss2, " in the ", 2)[1,2]
}
```

Score

```
df$PTS <-0
for (i in 1:nrow(df)) {
  ss1 <- str_split_fixed(df[i,2], "You are ", 2)[1,2]
  ss2 <- str_split_fixed(ss1, "by ", 2)[1,2]
  df$PTS[i] <- as.numeric(str_split_fixed(ss2, " points", 2)[1,1])
  if (grepl("down", ss1)) {
    df$PTS[i] <- df$PTS[i]*-1
  }
}
```

Final Dataset

Due to the fact there were only 9 cases of "Don't Know/it depends", and that it's not helpful, remove those cases.

```
df$SITUATION <- NULL
df <- subset(df, df$LABEL != "Don't know / it depends")
df$LABEL <- as.factor(df$LABEL)
df$DOWN <- as.factor(df$DOWN)
df$Q <- as.factor(df$Q)
head(df, 5)
```

```
## # A tibble: 5 x 7
##   LABEL             DOWN     YTG    YL   SEC Q        PTS
##   <fct>             <fct>  <dbl> <dbl> <dbl> <fct>  <dbl>
## 1 kick a field goal first     10    30     3 second    -3
## 2 kick a field goal second     0    45     3 second    -3
## 3 kick a field goal second     0    30     3 second    -3
## 4 kick a field goal second     0    45     3 fourth    -3
## 5 kick a field goal second     0    30     3 fourth    -3
```

Split into Training/Validation

While not shown here, distribution of labels between training and validation sets was assumed to be similar and was verified.

```
set.seed(74)
split_size <- floor(0.75*nrow(df))
split_index <- sample(seq_len(nrow(df)), size = split_size)

train <- df[split_index, ]
val <- df[-split_index, ]
```

Decision Trees

Bootstrap Aggregation aka *Belicheck Yourself*

```
tree1 <- bagging(LABEL~., data=train)
results1 <- table(predict(tree1, val[,2:7]), val$LABEL)
confusionMatrix(results1)
```

```
## Confusion Matrix and Statistics
##
##
##                      kick a field goal kneel down pass punt run
##   kick a field goal                113          1   11    6   6
##   kneel down                         1          8    0    1   0
##   pass                               8          5  222    0  28
##   punt                               3          2    7   52   2
##   run                                1          1   20    4 149
##
## Overall Statistics
##
##                Accuracy : 0.8356
##                  95% CI : (0.8049, 0.8633)
##     No Information Rate : 0.3994
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.7689
##
##  Mcnemar's Test P-Value : 0.02587
##
## Statistics by Class:
##
##                      Class: kick a field goal Class: kneel down Class: pass
## Sensitivity                            0.8968           0.47059      0.8538
## Specificity                            0.9543           0.99685      0.8951
## Pos Pred Value                         0.8248           0.80000      0.8441
## Neg Pred Value                         0.9747           0.98596      0.9021
## Prevalence                             0.1935           0.02611      0.3994
## Detection Rate                         0.1736           0.01229      0.3410
## Detection Prevalence                   0.2104           0.01536      0.4040
## Balanced Accuracy                      0.9256           0.73372      0.8745
##                      Class: punt Class: run
## Sensitivity              0.82540     0.8054
## Specificity              0.97619     0.9442
## Pos Pred Value           0.78788     0.8514
## Neg Pred Value           0.98120     0.9244
## Prevalence               0.09677     0.2842
## Detection Rate           0.07988     0.2289
## Detection Prevalence     0.10138     0.2688
## Balanced Accuracy        0.90079     0.8748
```

```
#varImp(tree1)
```

Random Forest aka *Andy Feed*

```
tree2 <- randomForest(LABEL~., data=train)
results2 <- table(predict(tree2, val[,2:7]), val$LABEL)
confusionMatrix(results2)
```

```
## Confusion Matrix and Statistics
##
##
##                      kick a field goal kneel down pass punt run
##   kick a field goal                119          2   12   11   7
##   kneel down                         0          7    0    0   0
##   pass                               5          5  226    0  29
##   punt                               2          2    8   50   1
##   run                                0          1   14    2 148
##
## Overall Statistics
##
##                Accuracy : 0.8449
##                  95% CI : (0.8147, 0.8718)
##     No Information Rate : 0.3994
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7817
##
##  Mcnemar's Test P-Value : 1.93e-05
##
## Statistics by Class:
##
##                      Class: kick a field goal Class: kneel down Class: pass
## Sensitivity                            0.9444           0.41176      0.8692
## Specificity                            0.9390           1.00000      0.9003
## Pos Pred Value                         0.7881           1.00000      0.8528
## Neg Pred Value                         0.9860           0.98447      0.9119
## Prevalence                             0.1935           0.02611      0.3994
## Detection Rate                         0.1828           0.01075      0.3472
## Detection Prevalence                   0.2320           0.01075      0.4071
## Balanced Accuracy                      0.9417           0.70588      0.8847
##                      Class: punt Class: run
## Sensitivity              0.79365     0.8000
## Specificity              0.97789     0.9635
## Pos Pred Value           0.79365     0.8970
## Neg Pred Value           0.97789     0.9239
## Prevalence               0.09677     0.2842
## Detection Rate           0.07680     0.2273
## Detection Prevalence     0.09677     0.2535
## Balanced Accuracy        0.88577     0.8818
```

```
#varImp(tree2)
```

Classification and Regression aka *Mike Shanahan-me-down*

```
tree3 <- rpart(LABEL~., data=train)
results3 <- table(predict(tree3, val[,2:7], type="class"), val$LABEL)
confusionMatrix(results3)
```

```
## Confusion Matrix and Statistics
##
##
##                      kick a field goal kneel down pass punt run
##   kick a field goal                112          3   21   13  18
##   kneel down                         0          0    0    0   0
##   pass                              14         11  178    0  29
##   punt                               0          3   25   50   2
##   run                                0          0   36    0 136
##
## Overall Statistics
##
##                Accuracy : 0.7312
##                  95% CI : (0.6954, 0.7649)
##     No Information Rate : 0.3994
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6272
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: kick a field goal Class: kneel down Class: pass
## Sensitivity                            0.8889           0.00000      0.6846
## Specificity                            0.8952           1.00000      0.8619
## Pos Pred Value                         0.6707               NaN      0.7672
## Neg Pred Value                         0.9711           0.97389      0.8043
## Prevalence                             0.1935           0.02611      0.3994
## Detection Rate                         0.1720           0.00000      0.2734
## Detection Prevalence                   0.2565           0.00000      0.3564
## Balanced Accuracy                      0.8921           0.50000      0.7733
##                      Class: punt Class: run
## Sensitivity              0.79365     0.7351
## Specificity              0.94898     0.9227
## Pos Pred Value           0.62500     0.7907
## Neg Pred Value           0.97723     0.8977
## Prevalence               0.09677     0.2842
## Detection Rate           0.07680     0.2089
## Detection Prevalence     0.12289     0.2642
## Balanced Accuracy        0.87132     0.8289
```

```
#varImp(tree3)
```

Conclusion

Feature Importance can also be evaluated. Similar to *Guess Who?*, feature importance describes the most efficient path of questioning to get to an answer. In this context, importance reveals which factors of the current play/game will lead to a play calling decision in the least amount of steps. In this experiment, the features ranked in descending order of importance are:

1. What down it is
2. Yards to Go
3. Yardline (position on field)
4. Score differential
5. Seconds remaining in the quarter
6. Which quarter it is

What I'm trying to say is... wait let me verify...

```
i <- nrow(val)+1
val[i, 2] <- as.factor('second') #second down
val[i, 3] <- as.numeric(1) #1 yard to go
val[i, 4] <- as.numeric(99) #99 yard line
val[i, 5] <- as.numeric(26) #26s on the clock
val[i, 6] <- as.factor('fourth') #fourth quater
val[i, 7] <- as.numeric(-4) #down by 4

predict(tree1, val[i,2:7])
```

```
## [1] kick a field goal
## Levels: kick a field goal
```

```
predict(tree2,val[i,2:7])
```

```
##                 1
## kick a field goal
## Levels: kick a field goal kneel down pass punt run
```

```
predict(tree3, val[i,2:7], type="class")
```

```
##                 1
## kick a field goal
## Levels: kick a field goal kneel down pass punt run
```

The Seahawks should have *kicked* the ball? Well nobody's perfect...